NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

THE SYSTEMS TEST ARCHITECT: ENABLING THE LEAP FROM TESTABLE TO TESTED

by

Javier A. Rinaldi

September 2016

Thesis Advisor: Oleg Yakimenko
Second Reader: Kristin Giammarco

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE THE SYSTEMS TEST ARCHITECT: ENABLING THE LEAP FROM TESTABLE TO TESTED			**5. FUNDING NUMBERS**
6. AUTHOR(S) Javier A. Rinaldi			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Test and evaluation is one of the cornerstones of the systems engineering process. Not only is it the main vehicle to obtain information about the adequacy of a system design, but it positively influences design decisions and the systems engineering process, if used from the earliest life-cycle stages. Based on its value, the Department of Defense (DOD) and industry both have placed an emphasis on a "shift-left" mentality approach in recent years. Despite this, little guidance or policy is available on how to achieve this mentality within the scope of the systems engineering process.

Through the analysis of the documented roles of test and evaluation in systems engineering, this thesis examines the concept that test and evaluation, based on its desired early involvement in the system engineering process, is a stakeholder in that process. In order to participate in that process, test and evaluation as an activity requires a proxy, which this work refers to as the "systems test architect." The conclusion is that the Systems Test Architect will positively influence the systems engineering process by becoming the proxy stakeholder for test and evaluation.

14. SUBJECT TERMS			**15. NUMBER OF PAGES**
test and evaluation, test architect			87
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

THE SYSTEMS TEST ARCHITECT: ENABLING THE LEAP FROM TESTABLE TO TESTED

Javier A. Rinaldi
Civilian, Department of the Navy
B.S., University of Puerto Rico, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2016

Approved by: Oleg Yakimenko, PhD
Thesis Advisor

Kristin Giammarco, PhD
Second Reader

Ronald Giachetti, PhD
Chair, Department of System Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Test and evaluation is one of the cornerstones of the systems engineering process. Not only is it the main vehicle to obtain information about the adequacy of a system design, but it positively influences design decisions and the systems engineering process, if used from the earliest life-cycle stages. Based on its value, the Department of Defense (DOD) and industry both have placed an emphasis on a “shift-left” mentality approach in recent years. Despite this, little guidance or policy is available on how to achieve this mentality within the scope of the systems engineering process.

Through the analysis of the documented roles of test and evaluation in systems engineering, this thesis examines the concept that test and evaluation, based on its desired early involvement in the system engineering process, is a stakeholder in that process. In order to participate in that process, test and evaluation as an activity requires a proxy, which this work refers to as the “systems test architect.” The conclusion is that the Systems Test Architect will positively influence the systems engineering process by becoming the proxy stakeholder for test and evaluation.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ACAT	acquisition category
CI	configuration item
DAG	Defense Acquisition Guidebook
DAU	Defense Acquisition University
DOD	Department Of Defense
DOT&E	director of operational test and evaluation
DT&E	developmental test and evaluation
GAO	government accountability office
INCOSE	international council on systems engineering
IPT	integrated product team
KPP	key performance parameter
M&S	modeling and simulation
MDA	milestone decision authority
MOE	measure of effectiveness
MOP	measure of performance
NDIA	National Defense Industrial Association
OT&E	operational test and evaluation
OTA	operational test activity
PM	program manager
PSM	practical software and systems measurement
RAM	reliability and maintainability
SA	systems architect
SE	systems engineer
SEBoK	Systems Engineering Body of Knowledge
SEP	systems engineering plan
SETR	systems engineering technical review
STA	systems test architect
T&E	test and evaluation
TEMG	test and evaluation management guide

TEMP	test and evaluation master plan
TMRR	technology maturation and risk reduction
TP	technical product
TPM	technical performance measure
US	United States
USD(AT&L)	Under Secretary Of Defense For Acquisition, Technology, And Logistics
WIPT	working-level integrated product team

EXECUTIVE SUMMARY

The process of test and evaluation is a critical component of any system development effort. Test and evaluation activities are where system performance and behavior is quantified, verified and validated. It is also where the discovery of the majority of errors and deficiencies in requirements and design occurs. Commonly, systems engineering models indicate test and evaluation activities as part of the verification and validation activities. With testing occurring so late in the development, correction of deficiencies becomes a costly endeavor. This observation has led to shift in mentality in Department of Defense policy that endeavors to integrate test and evaluation activities earlier in the process.

Outside of the Department of Defense policy, very little guidance details the representation, influence and interactions of the test and evaluation workforce with the systems engineering process. The Department of Defense policy that does exist is written for the acquisition of congressionally-approved programs of record. No model exists that describes how to achieve this early integration of test and evaluation for the systems engineering process in general.

In the current systems engineering management paradigm, the program manager is ultimately responsible for the development and execution of the test and evaluation strategy. It is not uncommon to program managers to have a less than ideal view of and relationship with the test and evaluation process and workforce. This contentious relationship and the lack of documented guidance on how to achieve the early integration of test and evaluation leads to a system engineering management deficiency whereby the program manager cannot reap the benefits of a thorough and rigorous test and evaluation strategy. A potential solution to this deficiency is to establish the role of the systems test architect, a test and evaluation domain expert responsible for the development and execution of the test and evaluation strategy.

The purpose of this thesis is to provide a model for the role of the systems test architect based on the documented roles and responsibilities of the test and evaluation

workforce within the systems engineering process. The potential benefit of such a role would be based on the increased probability of identifying and correcting deficiencies early in the system development effort. To develop the model for the systems test architect the following research question was explored.

Where and how does the role of the systems test architect intersect with the other roles in the system engineering process, based upon their respective literary descriptions?

The main role of the systems test architect is that of the “systems thinker” (Brewer, Emmert, and Guise 2012) for test and evaluation. The systems test architect takes a holistic view of test and evaluation, ensuring that test and evaluation is integrated throughout the system life cycle. The systems test architect is a coordinator, enabler, advocate and liaison between the program manager, systems engineering team and the test evaluation workforce. The systems test architect is the advocate for test and evaluation, ensuring that throughout the system development effort, test and evaluation activities engage in and influence the process. The system test architect is the steward of the test and evaluation strategy, providing expertise, information and clarity regarding test and evaluation needs, events and results during program, technical and readiness reviews. The systems test architect is the advisor for matters of test and evaluation, providing a domain-expert enable lens for requirements and architecture definitions. The systems test architect can be summarized as being the proxy stakeholder for test and evaluation. As a process, test and evaluation has no voice or influence. With the systems test architect as stakeholder, test and evaluation gains influence over the entirety of the systems development effort.

As the party responsible for test and evaluation, the systems test architect develops and executes the tapestry of activities that encompass the test and evaluation strategy. This entails influencing design for testability, developing test plans and evaluation methods, coordinating test schedules and test resources. The systems test architect coordinates across the test and evaluation strategy to develop a test strategy that covers the entirety of the system requirements in a rigorous, thorough, repeatable and statistically-defensible approach. The systems test architect interacts with program

managers, systems engineers, developers, testers and evaluators to develop and execute the test strategy.

With the establishment of the role of systems test architect, the system should reap the benefits of early integration of the test and evaluation activities. Among these benefits would be improved systems engineering processes, improved requirements definitions, improved communications, reduced technical risks and reduced life-cycle costs. The program manager should weigh these potential benefits against the possible costs of addressing system deficiencies.

Based on the uniqueness of this role, future efforts based on this research should seek to quantify the value of a role like the systems test architect and exploring the potential areas where systems engineering management lacks guidance which prevents the systems engineering process from delivering on its goal of delivering good systems.

References

Brewer, James, Louisa Guise, and Jerry Emmert. 2012. “The Value of the Test Architect.” Paper presented at the NDIA 28th Annual National Test and Evaluation Conference, Hilton Head, SC, March 20.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to acknowledge one of the greatest influences in my life, my grandfather, Manolo. His example of perseverance through hard work and his ability to reinvent himself and succeed despite great odds is a continual source of inspiration in my life. I would also like to thank my wife, Erin, for her patience and understanding through this process, and for listening to all my outlandish ideas about “walls” and such. I would also give my thanks to my son, for being my greatest source of joy and inspiring me to become a greater role model.

I would like to thank my professors and the staff at the Naval Postgraduate School, for opening my mind to systems thinking and pushing me beyond my limits to perform what I once though unlikely.

Finally, I would like to thank my fellow 151 cohorts, for making these two years together an unforgettable experience. I wish you all the greatest success and joy in life.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Testing, in its many forms, is one of the most important and time-consuming processes in the development of a system. It is during testing that system behavior is quantified, verified and validated; it is also the process in which most errors in design and requirements are discovered. In most systems engineering models, testing occurs mainly as a function of the verification and validation phases of the systems engineering process. When testing discovers issues with the system design, the correction of those errors can be costlier to fix than if they were discovered during an earlier stage (United States General Accountability Office 2000). The need to involve the test and evaluation (T&E) workforce in the acquisition process has become a focus of the Department of Defense (DOD) in recent updates to the *Defense Acquisition Guidebook* (DAG) and the overarching acquisition document, DOD Instruction 5000.02. These updates are influenced by textbooks (Grady [2010]), guides, and materials that also expose the potential benefits of this early involvement of the test and evaluation workforce.

Despite its espoused importance, very little guidance details the representation, influence and interactions of the test and evaluation workforce with the systems engineering process. What guidance does exist takes the form of Department of Defense instructions and guidance (namely, Army Test and Evaluation Command [2003], USD(AT&L) [2015], and Defense Acquisition University [2012]) and is written for the management of large acquisitions of military systems and not in the general context of systems engineering. The International Council on Systems Engineering (Roedler and Jones 2003) provides a description of the influences and value of the test and evaluation involvement; however, unlike the DOD guidance, it does not apportion the responsibility for test and evaluation involvement to any representative.

In recent years, some industry bodies have begun to allocate the responsibility for the management of the test and evaluation program to systems engineering interaction (Brewer, Guise, and Emmert 2012; Morrison 2007; Page 2008) with a role named the

systems test architect (Brewer, Guise, and Emmert 2012; Guise 2012; Manas 2014; Manas and Wilson 2014). They have defined some of the functions, responsibilities, influences and benefits of that role; however, a model for this role in the context of systems engineering is not readily available.

B. PROBLEM STATEMENT

There is a deficiency in the available guidance as to how to implement the integration of test and evaluation in the systems engineering process. Despite the well-documented benefits of a rigorous test and evaluation program, it is apparent that few sources outside the DOD are tackling this integration. And in the case of the DOD, the policies and guidance are tailored to the acquisition of large-budget programs of record. Smaller bodies that desire to integrate test and evaluation into their processes must devise their own approach. As it stands, the program manager is solely responsible for the development and execution of the test and evaluation plan, since no other party has this role. However, the program manager can delegate this responsibility as needed.

C. PURPOSE

The purpose of this research is to provide an overview of the available material on the interactions, roles and responsibilities of the test and evaluation workforce in the systems engineering process. The use of those materials allows for the extraction of the test and evaluation functions that add value to the systems engineering process. Those functions are the roles and responsibilities that a systems test architect would embrace to the benefit the systems engineering process.

D. RESEARCH QUESTION

The focus of this research is to use the following set of questions to act as a lens through which to read and dissect the research material and extract the model for the roles and responsibilities of the systems test architect. By condensing the information from the research in this body of work, a business case will emerge for the establishment and inclusion of this role into the systems engineering process. The research question is:

Where and how does the role of the systems test architect intersect with the other roles in the system engineering process, based upon their respective literary descriptions?

The main research question is broken down into several smaller questions:

a. What is the role of the systems test architect within the systems engineering process?

b. How does the systems test architect interact with the other roles within the systems engineering process?

c. How does the systems test architect improve the systems engineering process?

d. Should the role of the systems test architect be more thoroughly integrated (and formally defined) into systems engineering?

e. How does the systems test architect affect the cost and quality of a project?

E. POTENTIAL BENEFIT OF STUDY

Many sources credit the test and evaluation activities with the discovery of most system design defects, and also state that the correction of those defects upon arrival at the verification and validation phase is costlier than correction at an earlier phase. In light of these findings, the DOD and other industry bodies have begun a move towards a "shift-left" mentality (United States General Accountability Office 2000; Manas 2014) when it comes to the involvement of test and evaluation in the system acquisition cycle.

The goal of this work is to provide a basic framework for the involvement of the systems test architect as a liaison and representative for test and evaluation in the systems engineering process and steward for the test and evaluation program. It will be evident that this role will facilitate the "shift-left" mentality and continue the tradition of systems engineering as a process that enables building better, more complete systems.

F. SCOPE

The focus of this research is to identify and tie together, from available materials, the roles, responsibilities, interactions and value added by the systems test architect. This research is not an argument on the value of test and evaluation in systems engineering, that has been covered by multiple sources (Bodmer 2003; Barret 2009; United States General Accountability Office 2000). Rather, examines the concept that test and

evaluation, based on its desired early involvement in the system engineering process, is a stakeholder in that process. In order to participate in that process, test and evaluation as an activity requires a proxy, which this work refers to as the Systems Test Architect. The resulting hypothesis is that the Systems Test Architect will positively influence the systems engineering process by becoming the proxy stakeholder for test and evaluation.

The remainder of this research is divided into the following sections:

- Chapter II is a review of the general involvement of test and evaluation in the systems engineering process. This provides context on the roles and value of test and evaluation to systems engineering.

- Chapter III is a review of the currently available guidance on the representation of the test and evaluation workforce in the systems engineering process. This provides context on the handling of test and evaluation activities, and what roles they play in systems engineering.

- Chapter IV is a discussion of the fit of the systems test architect, covering the potential set of roles and responsibilities of the systems test architect, and the interactions between the systems test architect and the systems engineering process. This forms the model for the role of systems test architect.

- Chapter V is a discussion of the expected value of the systems test architect in the systems engineering process. Given the amalgamation of roles in Chapter IV, this chapter explores the potential values that a program reaps from the systems test architect. Chapter V also presents a fictional scenario that exposes the potential benefits of the systems test architect.

- Chapter VI is a brief synopsis of the research in the form a discussion of the research questions, as well as conclusions and topics for further study.

G. METHODOLOGY

This research was conducted using the following methodology:

- Conduct a literature review of instruction materials, DOD and industry guidance, policy, reports and presentations related to the involvement of test and evaluation in systems acquisition and development.

- Provide an analysis of curated list of sources from literature review and extraction of relevant best practices, organizational and procedural recommendations.

- Develop a model for the role of systems test architect.

II. TEST AND EVALUATION IN SYSTEMS ENGINEERING

This chapter provides a discussion of the overall process of systems engineering with particular emphasis on the involvement of test and evaluation at the different stages of the process. To aid in this description, test and evaluation is defined as an integral set of activities, where testing can take the following forms (United States Department of Defense 2001, 66; Buede 2009, 41):

- Analysis is the use of mathematical models or simulation techniques to calculate a required parameter.
- Inspection is the use of human interaction (e.g., vision, touch) to ascertain the value of a parameter.
- Demonstration is the operation of an individual component or subsystem in a limited scope and environment to ascertain a behavior or capability.
- Instrumented Test is the operation of an individual component or subsystem with the use of measurement instrumentation to obtain quantified values for parameters of interest.

Evaluation then takes the form of analysis of the test results, the formation of a judgement based on those results and the formation of a recommendation should a correction be required.

A. DEFINITIONS

- *System*: a system is a set of interrelated components functioning together towards some common objective(s) or purpose(s) (Blanchard and Fabrycky 2011).
- *Systems Engineering*: systems engineering is an interdisciplinary approach and means to enable the realization of successful systems (INCOSE 2016b).
- *Architecture*: the structure—in terms of components, connections, and constraints—of a product, process, or element (Maier and Rechtin 2000).
- *Architect*: a person who designs and guides a plan or undertaking (Merriam-Webster 2016).

- *Systems Architecting*: The art and science of creating and building complex systems. That part of systems development most concerned with scoping, structuring, and certification (Maier and Rechtin 2000).

The Test and Evaluation Management Guide (United States Department of Defense 2012) defines the following parameters:

- *Test*: test denotes any program or procedure that is designed to obtain, verify, or provide data for the evaluation of any of the following: (1) progress in accomplishing developmental objectives; (2) the performance, operational capability, and suitability of systems, subsystems, components, and equipment items; and (3) the vulnerability and lethality of systems, subsystems, components, and equipment items

- *Evaluation*: evaluation denotes the process whereby data is logically assembled, analyzed, and compared to the expected performance to aid in systematic decision making. It may involve review and analysis of qualitative or quantitative data obtained from design reviews, hardware inspections, modeling and simulation, hardware and software testing, metrics review, and operational usage of equipment

- *Test and Evaluation*: a process by which a system or components are tested and results analyzed to provide performance related information

B. GENERAL THEORY OF SYSTEMS ENGINEERING

The definition of systems engineering has been rewritten many times by its practitioners (Blanchard and Fabrycky 2011, 17), influenced by scope, experience and interpretation of the processes that make up systems engineering. Despite a myriad of definitions, most definitions share the following attributes:

- Hierarchical: systems are developed in a sequence of steps, increasing in detail, with each step informed or influenced by its predecessor.

- Iterative: the steps in the system development process are subject to rework or reinterpretation given feedback from subsequent steps (Blanchard and Fabrycky 2011, 100).

- Life cycle–oriented: the entirety of the systems’ life-cycle stages, from design to disposal, is taken into consideration during the development effort.

- Interdisciplinary: the development of a system requires a team of professionals with differing sets of knowledge to cover all the design objectives (Blanchard and Fabrycky 2011, 15,114; Grady 2010, 34; Maier and Rechtin 2000, 8–9).

- Front-loaded: systems engineering derives a lot of its value based on the practice of making positively impactful decision early in the system development effort. With the knowledge that a negatively impactful finding later in the development process is factors of magnitude costlier to correct (Buede 2009, 33; Blanchard and Fabrycky 2011, 48).

The general approach of systems engineering is one of translation of user needs to a delivered system that meets those needs by segmenting the development effort into a structured and informed set of steps. Aside from the common attributes listed previously, and its structured approach; systems engineering derives a lot of its value from being a well-documented process. At every stage, requirements, plans, decisions, relevant data and the like are documented and tracked. This is not only of value to the process at hand but also to future endeavors that may benefit from the lessons learned (for good or bad) during one evolution.

When it comes to test and evaluation in systems engineering, the intent is to include it early in the process. Unfortunately, there is little guidance on how program manager and system engineer achieve this integration. As such test and evaluation activities end up having interactions with several members of the system engineering team and cooperation suffers. Figure 1 shows some of these relations in a small scope.

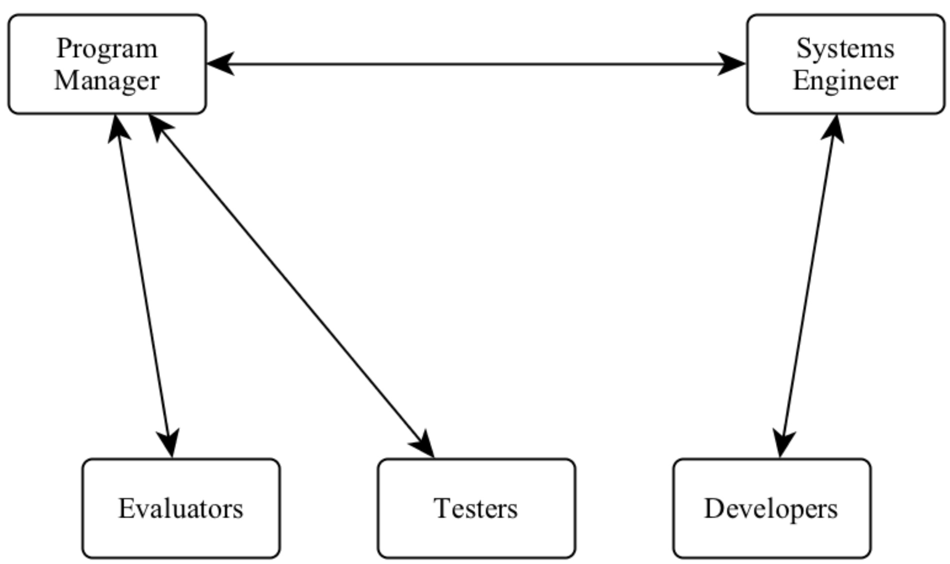

Figure 1. Test and Evaluation Interaction Map

C. A TEST AND EVALUATION VIEW OF THE SYSTEMS ENGINEERING PROCESS MODEL

The systems engineering process is usually represented as a model; that is, it is depicted as a graphical representation of the steps required to achieve the final product. This allows the systems engineer to substantiate the transformation from user needs to a finished system. A plethora of models have been developed over decades of the evolution of systems engineering (Blanchard and Fabrycky 2011, 36–37), tailored to different realms of application. This discussion uses the “Vee” process model (Figure 2) to highlight the stages of the systems development process. This particular model has become ubiquitous in the practice on systems engineering within the Department of Defense as well as industry and academia.

The roles and relationships detailed here are by no means all-inclusive; they are presented without presumption as to how they are put into practice. The goal is for the reader to acquire a general understanding of the subject matter and bring into focus the areas in this process where test and evaluation intersect.

In the current paradigm, the test and evaluation program is the responsibility of the program manager. Beyond that, there is little or no clarification on who is responsible for the different aspects of the test and evaluation program. This leads to a disconnect, where the systems engineer is working to incorporate test and evaluation, yet has little interaction with those responsible for executing the program.

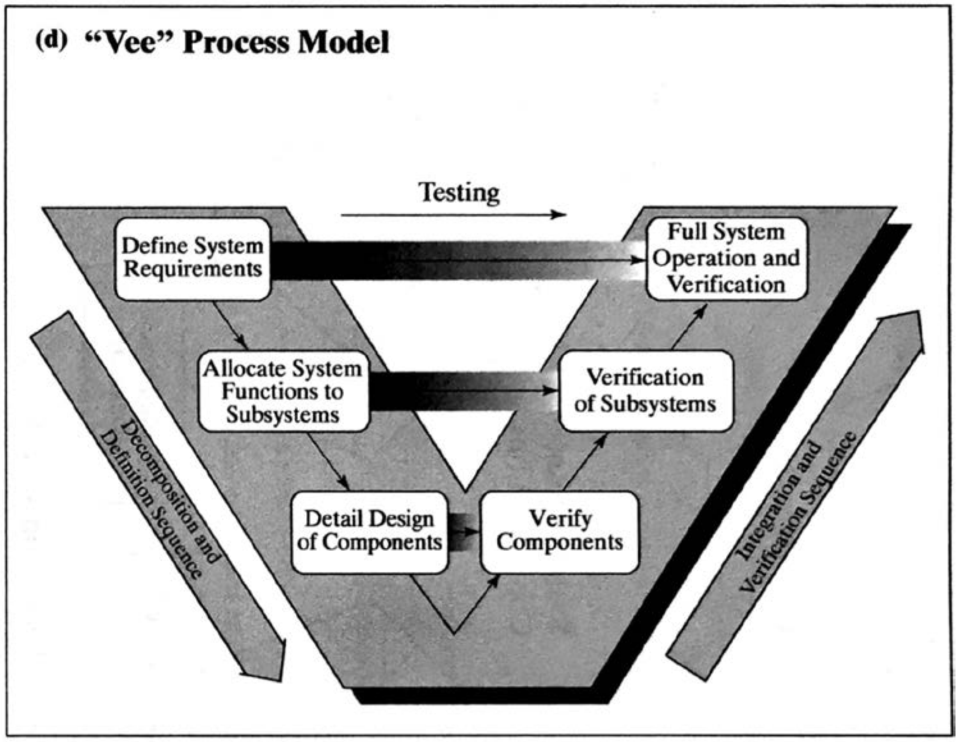

Figure 2. Systems Engineering "Vee" Process Model. Source: Blanchard and Fabrycky (2011).

As previously stated, the systems engineering process requires an interdisciplinary approach, involving both technical and managerial disciplines applied to the synthesis and integration of a system, taking into consideration the entirety of the systems' life cycle. Systems engineers combine components to enable functions, in support of capabilities stated as requirements driven by needs. Systems engineering is both a top-down and bottom up approach to system design, and a coordination of efforts, processes and people to achieve the goal of a "good" system. A "good" system would be defined as one that fulfils the need of the user or customer and behaves they expect.

1. Definition of the System Requirements

A system has a purpose, this purpose can be seen as the need of the customer, who is the main stakeholder in the process. This need is translated by the systems engineering team into a set of system requirements.

In this process, the systems engineer takes the stated need of the user and the constraints imposed upon by other stakeholders and produces a set of high-level requirements. A common name for these high-level requirements is Measures of Effectiveness (MOE); these are the most abstract set of requirements and their measure relates directly to how well the system fulfils the customer need. Next, the systems engineer refines the high-level requirements into more specific requirements and Measures of Performance (MOP) related to the function, design and performance of the system. Some requirements require special attention throughout the systems engineering process, these are tracked over time to make informed decisions at key milestones, ensure that goals are being met and manage risk; these special requirements are Technical Performance Measures (TPM). At this stage, the systems engineer should also be generating test scenarios for these requirements.

Requirements definition is a critical step in the systems engineering process, given that the decisions made during this stage have consequences for entirety of the systems' life cycle. Ensuring good requirements requires great attention to detail. Grady (2010, 277–278) details seven attributes for good requirements, detailed as follows:

1. Traceable – the hierarchy of every requirement can be traced to a driving need
2. Correct in style – every requirement should be properly expressed in a language matching the customers'
3. Understandable – every requirement should be clearly and briefly stated in unambiguous language
4. Single in purpose – every requirement should stand on its own; that is, related or coupled requirements should be de-coupled
5. Quantifiable – requirements should be stated, as often as possible, in a quantified manner
6. Verifiable – every requirement should be verifiable through a practical process
7. Sensible – every requirement should "make sense," insofar that a requirement cannot break the laws of nature or be beyond the reach of attainable technology.

The quantifiable, verifiable and sensible attributes are of particular interest to the test and evaluation workforce, as they influence the verification and validation approach and requirements. Buede (2009, 344–345) makes a similar argument regarding requirements language and attribute, stating that, “In order for verification to be successful, the originating and derived requirements must be testable; that is, the requirements must be single statements that unambiguous, understandable and verifiable.” Other sources like the United States Department of Defense (2001) and Blanchard and Fabrycky (2001) also argue for these attributes in requirements definition.

2. Functional Decomposition and Allocation (Functional Architecting)

Once the system requirements have been sufficiently refined, the systems engineer or architect will establish the functions that will fulfill those requirements. This process requires a thorough understanding of the system purpose. The systems engineer or architect will arrange and rearrange hierarchies of functions, stated as verb-noun sets, into an ever more detail hierarchy of functions. The systems engineer and systems architect perform this decomposition while maintaining a solution neutral architecture definition.

The result of the process of the functional decomposition is the creation of the functional architecture, which is a specific arrangement of the functional decomposition hierarchy. The process of creating a functional architecture can lead to many possible arrangements, it is up to the systems engineer or architect to envision the best possible arrangement.

During the functional decomposition process the systems engineer or architect may engage the test and evaluation workforce to produce models or simulations of the system of interest and refine the functions that will achieve the desired system behavior.

3. Design Synthesis (Physical Architecting)

The design synthesis process entails the translation of the functional decomposition into possible sets of physical architectures. Although separate from the functional decomposition in this description, the process of design synthesis and the

identification of possible physical architectures are symbiotic with the functional architecting process. As such, the systems engineer or architect performs these two processes simultaneously. The functional architecture may define what a component or subsystem must do and the physical architecture will attempt to fit that requirement, if the requirement should not be feasible then the functional architecture must change to allow a physical architecture to emerge that will meet the required behavior.

During the design synthesis process, the systems engineer or architect may engage the test and evaluation workforce to produce models or simulations of components, component arrangements or sub-systems to ensure the system will achieve the desired functions. The test and evaluation workforce may also be engaged to support the generation of trade studies, as these may require testing beyond the capability of modeling or simulation tools.

4. Design Implementation

The design implementation process begins the integration activities in the systems engineering process, usually depicted beginning at and encompassing the right side of the "Vee" model. This process encompasses the fabrication, assembly or coding of the desired system components or subassemblies into a structure that matches the defined physical architecture. This process may require fabrication of prototypes to ensure system performance and behavior, which may affect the possible physical architectures. Here as well there is a feedback loop to the design synthesis step. Inverse to the top-down approach that is taken during the design and decomposition portion of the "Vee" model, the integration activities are carried out bottom-up. The developers build up the system from the most basic components to the entirety of the system. All the while, the test and evaluation workforce is testing the system to ensure that requirements are satisfied, the results and tester feedback is given to the developers to influence the design.

During the design implementation process, the test and evaluation workforce should be heavily involved in the preliminary testing of assembled components and prototypes. Testing is the keystone activity in the integration process, as Buede (2009, 4) states, "Integration brings all of the detailed elements of the overall design together

through a process of testing (or qualification) to achieve a valid system for meeting the needs of the stakeholders." As such, the voice of the tester should be loud and clear in the ear of the developers.

5. Component Verification

During the component verification phase, the individual components and sub-systems are assembled and tested against the requirements set forth during the requirements definition and subsequent phases. This verification of the components is the reason for the need of clear, quantifiable and testable requirements. For every requirement there should be a desired threshold of performance and a method by which to measure that parameter. A test and evaluation master plan documents this relationship between requirement and test for the entire development and life cycle of the system, including the details for each test scenario. As indicated as the beginning of this chapter, test and evaluation can take on many forms. Selection of the correct methodology is critical to ensure the correct data is collected and the proper conclusion reached on the adequacy of the test component. Critical requirements, the aforementioned technical performance measures, are of particular interest to the systems engineering team, developers and program managers.

At this stage the test and evolution workforce is fully involved; assembling and testing the system components as specified in different test plans. Some tests are more involved, requiring a high degree of planning and coordination between personnel, facilities and test assets. Other tests may be as simple as a bench test. Nevertheless, every test produces some data that the test and evaluation workforce will collect, analyze, and provide to the systems engineering team and evaluators to consume and influence their decisions.

6. System Verification

The process of system verification entails much the same as the component verification stage, but applied to the system as whole. In the context of verification, the system as whole is tested to ensure that it meet the requirements set forth at the beginning of the systems engineering process. As Buede (2009, 344) states, "Verification is the

matching of CIs [configuration items], components, subsystems and the system to their corresponding requirements to ensure that each has been built right."

At this stage, the test and evaluation workforce engages in performing systems level tests; however, these tests may or may not occur in the operational environment for the system. As at this stage the objective is to ensure the system was built right. This stage of testing usually requires a high degree of coordination and cooperation among designers, builders, testers and evaluators. This ensures that the systems engineering team captures any issues with the system design prior to moving to the next stage.

7. System Validation

In this stage, the system as a whole is tested as it was during the system verification stage; however, during the system validation, the objective is to ascertain the right system was built (Buede 2009, 342). As such, the testing here involves the intended operational environment, including users and any conditions that the customer requires. Because the intention at this stage is to demonstrate behavior over performance, the involvement of the test and evaluation workforce is usually limited in comparison to the previous stages.

D. CHAPTER SUMMARY

This chapter presented a summary of the systems engineering process, with emphasis on how test and evaluation is vital to that process. It is clear that test and evaluation is interwoven in the systems engineering process, at least in theory. The essence of systems engineering is a knowledge-driven effort; test and evaluation activities provide much of that data.

The next chapter is a summary and discussion of the DOD and industry policy and guidance discovered during literature review. Next chapter emphasizes any defined roles, responsibilities they may have, and how they interact with or influence the systems engineering process.

III. TEST AND EVALUATION POLICY AND GUIDANCE

This chapter presents a review of systems acquisition and systems engineering policy and guidance documents and resources that prescribe or describe the involvement of the test and evaluation workforce in the systems engineering process. With the potential impact of test and evaluation on project schedule, cost and the overall acceptance of a system, there is no shortage on guidance.

A. DEPARTMENT OF DEFENSE POLICY ON TEST AND EVALUATION

This section presents policy and guidance on test and evaluation between federal and DOD-wide levels, the policies specific to each branch of service are interpretations of the documents detailed here. An evaluation of this guidance reveals that these policies are established mainly for major acquisition programs, although individual activities may expand and reinterpret them to suit their development needs.

1. United Stated Title, Code 10 – Armed Forces

Title 10 of the United States code of laws is the guiding policy for the organization, roles and mission of each branch to the U.S. military. Title 10 establishes the definitions for Operational Test and Evaluation (OT&E) and Developmental Test and Evaluation (DT&E), as well as defining policy on the person or persons responsible for providing oversight over those activities:

a. Operational Test and Evaluation

Operational Test and Evaluation is the activity of planning and conducting field tests under realistic conditions of any component or sub-system with the objective of determining its effectiveness or suitability and the evaluation of the results of that activity. Title 10, §139 establishes the position of Director of Operational Test and Evaluation, appointed by the president of the Unites States, based on merit to perform the following duties:

- Provide advice and guidance to the Secretary of Defense and the Under Secretary of Defense for Acquisition, Technology, and Logistics (USD(AT&L)) on all matters relating to Operational Test and Evaluation.

- Develop policies and guidance relating to the conduct of Operational Test and Evaluation.

- Monitor and review all Operational Test and Evaluation within the Department of Defense.

- Coordinate cross-organizational Operational Test and Evaluation activities.

- Provide input and recommendations on investments in Operational Test and Evaluation workforce and infrastructure.

- Provide or delegate oversight of test events.

- Maintain and propagate Operational Test and Evaluation results.

The Director of Operational Test and Evaluation is not involved in Developmental Test and Evaluation activities, except to provide non-binding advice on matters related to test and evaluation.

b. Developmental Test and Evaluation

Developmental Test and Evaluation is defined as the activity of planning and conducting tests of a component, subsystem or assembly to establish conformance with requirements and specifications (e.g., contractual, technical performance, supportability and interoperability) in order to collect measurable data and evaluate of the results of that activity.

Title 10, §139b establishes the position of Deputy Assistant Secretary of Defense for Developmental Test and Evaluation, appointed by the Secretary of Defense, based on merit and experience in the practice of test and evaluation, to perform the following duties:

- Provide advice to the Secretary of Defense and the Under Secretary of Defense for Acquisition, Technology, and Logistics (USD(AT&L)) on all matters relating to Developmental Test and Evaluation.

- Coordinate with the Deputy Assistant Secretary of Defense for Systems Engineering to "ensure that the developmental test and evaluation

activities of the Department of Defense are fully integrated into and consistent with the systems engineering and development planning processes of the Department" (United States Code, Title 10, Armed Forces, Subtitle A, §139b 2015).

- Develop guidance and policy for the conduct of Developmental Test and Evaluation within the Department of Defense.
- Review the Test and Evaluation Master Plan (TEMP) for every major defense program.
- Monitor, review and advice Developmental Test and Evaluation activities to ensure the establishment of best practices.
- Serve as advocate for the Developmental Test and Evaluation workforce.
- Provide guidance on the investment in the Developmental Test and Evaluation workforce and infrastructure.
- Assess and monitor the technological maturity of emerging technologies being integrated into defense acquisition programs.

Title 10, §139b also establishes and briefly describes the position of Chief Developmental Tester, to assist the Deputy Assistant Secretary of Defense for Developmental Test and Evaluation on tasks related to Developmental Test and Evaluation for specific programs of record. The chief developmental tester has the following duties:

- Coordinate program planning, management and oversight of Developmental Test and Evaluation.
- Communicate with contractor and governmental activities performing Developmental Test and Evaluation within the program.
- Assist the program manager in making technical judgments of Developmental Test and Evaluation results.

The three roles defined by Title 10 provide an initial set of attributes for the systems test architect if they are considered outside the context of the Department of Defense. These roles clearly show that earlier and more thorough integration of test and evaluation requires a responsible party to influence the decisionmakers and the process.

2. Department of Defense Directive 5000.01

Department of Defense directive 5000.01 is the top level systems acquisition guidance, prepared by the Under Secretary of Defense for Acquisition, Technology, and Logistics (USD(AT&L)) for any system under the Department of Defense. It establishes the Defense Acquisition System and its goal is to ensure the proper management of the investments in technologies, programs and products for national defense in the present and future. The primary objective of this document and its derived guidance is "to acquire quality products that satisfy user needs with measurable improvements to mission capability and operational support, in a timely manner, and at a fair and reasonable price" (USD(AT&L) 2003).

Directive 5000.01 establishes that programs shall use and adapt technologies and practices that reduce costs and encourage teamwork. Programs shall ensure collaboration across the spectrum of stakeholders "Teaming among warfighters, users, developers, acquirers, technologists, testers, budgeters, and sustainers shall begin during capability needs definition" (USD(AT&L) 2003). Furthermore, programs shall integrate test and evaluation throughout the acquisition process to provide essential decision-making information, asses technical performance parameters (TPM) and program viability.

3. Department of Defense Directive 5000.02

Department of Defense directive 5000.02 provides a higher level of detail to the policies and guidance set forth in 5000.01. It defines the process and management of defense acquisition, from identification of need to disposal. Throughout this directive, test and evaluation is interwoven with the systems engineering process that results in the development of a system. The use and value of test and evaluation is reiterated continuously throughout the directive.

Directive 5000.02 establishes the hierarchy of acquisition categories (ACAT) for acquisition programs, dependent on level of funding, complexity or request by the Milestone Decision Authority (Figure 3). These categories carry with them escalating level of oversight, with ACAT I being the most restrictive. 5000.02 defines the basic phases of the acquisition process and decision milestones with associated reviews

(Figure 4), which ensure program readiness to proceed to subsequent phases. The Milestone Decision Authority and Program Manager tailor these phases into different acquisition models as needed to fit the program needs. These program models also establish the phases that correspond to Developmental Test and Evaluation and Operational Test and Evaluation and the corresponding activities they support.

ACAT	Reason for ACAT Designation	Decision Authority
ACAT I	• MDAP (10 U.S.C. 2430 (Reference (g))) ○ Dollar value for all increments of the program: estimated by the DAE to require an eventual total expenditure for research, development, and test and evaluation (RDT&E) of more than $480 million in Fiscal Year (FY) 2014 constant dollars or, for procurement, of more than $2.79 billion in FY 2014 constant dollars ○ MDA designation • MDA designation as special interest[1]	ACAT ID: DAE or as delegated ACAT IC: Head of the DoD Component or, if delegated, the CAE (not further delegable)
ACAT IA[2, 3]	• MAIS (10 U.S.C. 2445a (Reference (g))): A DoD acquisition program for an Automated Information System[4] (AIS) (either as a product or a service[5]) that is either: ○ Designated by the MDA as a MAIS program; or ○ Estimated to exceed: ▪ $40 million in FY 2014 constant dollars for all expenditures, for all increments, regardless of the appropriation or fund source, directly related to the AIS definition, design, development, deployment, and sustainment, and incurred in any single fiscal year; or ▪ $165 million in FY 2014 constant dollars for all expenditures, for all increments, regardless of the appropriation or fund source, directly related to the AIS definition, design, development, and deployment, and incurred from the beginning of the Materiel Solution Analysis Phase through deployment at all sites; or ▪ $520 million in FY 2014 constant dollars for all expenditures, for all increments, regardless of the appropriation or fund source, directly related to the AIS definition, design, development, deployment, operations and maintenance, and incurred from the beginning of the Materiel Solution Analysis Phase through sustainment for the estimated useful life of the system. • MDA designation as special interest[1]	ACAT IAM: DAE or as delegated ACAT IAC: Head of the DoD Component or, if delegated, the CAE (not further delegable)
ACAT II	• Does not meet criteria for ACAT I or IA • Major system (10 U.S.C. 2302d (Reference (g))) ○ Dollar value: estimated by the DoD Component head to require an eventual total expenditure for RDT&E of more than $185 million in FY 2014 constant dollars, or for procurement of more than $835 million in FY 2014 constant dollars ○ MDA designation[5] (10 U.S.C. 2302 (Reference (g)))	CAE or the individual designated by the CAE[6]
ACAT III	• Does not meet criteria for ACAT II or above • An AIS program that is not a MAIS program	Designated by the CAE[6]

1. The Special Interest designation is typically based on one or more of the following factors: technological complexity; congressional interest; a large commitment of resources; or the program is critical to the achievement of a capability or set of capabilities, part of a system of systems, or a joint program. Programs that already meet the MDAP and MAIS thresholds cannot be designated as Special Interest.
2. When a MAIS program also meets the definition of an MDAP, the DAE will be the MDA unless delegated to a DoD Component or other official. The DAE will designate the program as either a MAIS or an MDAP, and the Program Manager will manage the program consistent with the designation.
3. The MDA (either the DAE or, if delegated, the DoD Chief Information Officer (CIO) or another designee) will designate MAIS programs as ACAT IAM or ACAT IAC. MAIS programs will not be designated as ACAT II.
4. AIS: A system of computer hardware, computer software, data or telecommunications that performs functions such as collecting, processing, storing, transmitting, and displaying information. Excluded are computer resources, both hardware and software, that are an integral part of a weapon or weapon system; used for highly sensitive classified programs (as determined by the Secretary of Defense); used for other highly sensitive information technology (IT) programs (as determined by the DoD CIO); or determined by the DAE or designee to be better overseen as a non-AIS program (e.g., a program with a low ratio of RDT&E funding to total program acquisition costs or that requires significant hardware development).
5. When determined by the USD(AT&L) (or designee), IT services programs that achieve the MAIS threshold will follow the procedures applicable to MAIS programs specified in this instruction. All other acquisitions of services will comply with Enclosure 9 of DoD Instruction 5000.02 (Reference (h)) until cancelled by issuance of the new acquisition of services instruction.
6. As delegated by the Secretary of Defense or Secretary of the Military Department.

Figure 3. Description and Decision Authority for Acquisition Categories (ACAT) I-III. Source: United States Department of Defense (2015)

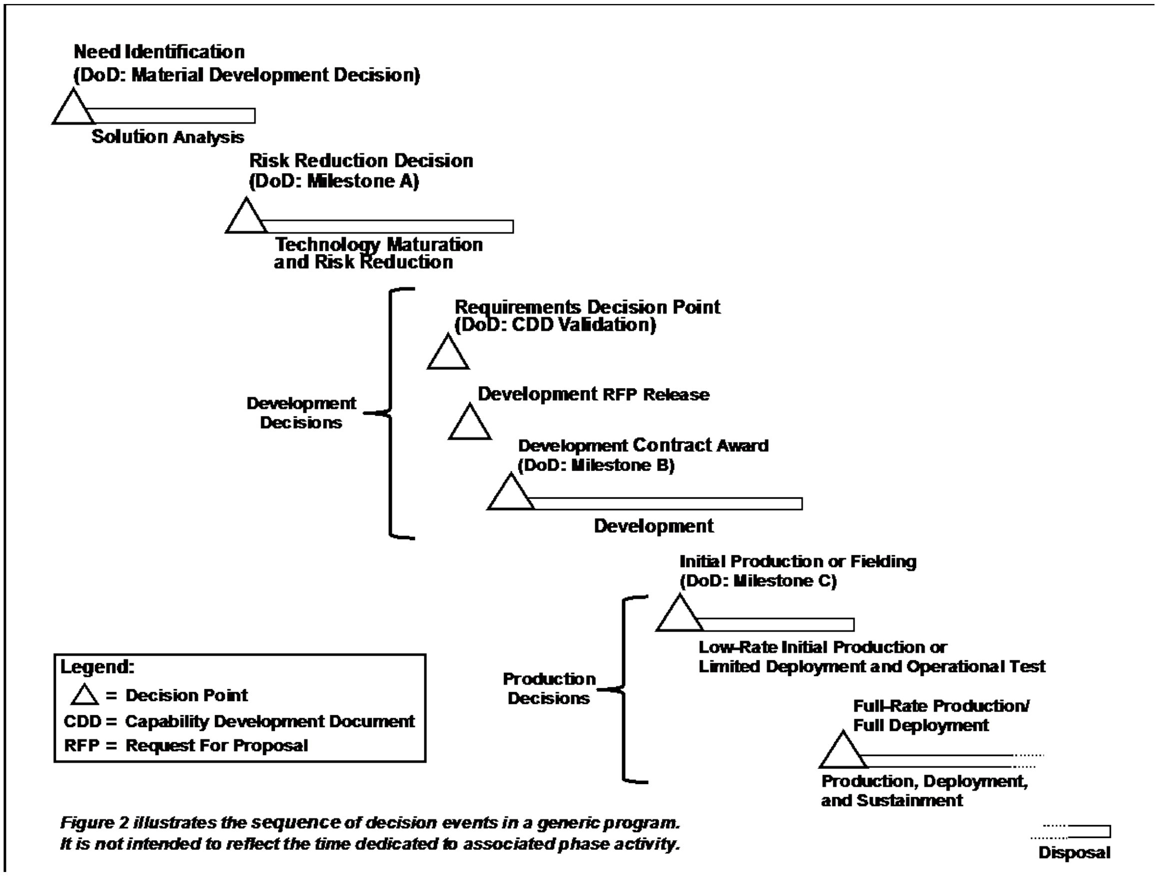

Figure 4. Generic Acquisition Phases and Decision Points. Source: United States Department of Defense (2015)

Directive 5000.02 defines some of the basic documentation requirements for program execution in support of decision-making and planning activities. Key among these documents are the Systems Engineering Plan (SEP), which is the primary management tool for systems engineering activities, and the Test and Evaluation Master Plan (TEMP), which is the primary management tool for the test and evaluation activities.

Directive 5000.02 provides some guidance on the planning of test and evaluation activities and the establishment of performance metrics beginning at the Technology Maturation and Risk Reduction (TMRR) phase, which precedes Milestone A. The use of test and evaluation activities during Milestone B to produce and test prototypes component and sub-systems to perform verification tests, trade analyses and risk

reduction studies. In 5000.02, the use of Developmental Test and Evaluation starts at Milestone B to provide feedback to the program manager on system performance and requirements fulfilment.

Directive 5000.02 emphasizes the cross-collaboration between Developmental and Operation Test and Evaluation activities to reduce total required tests by coordinating activities and test assets or sharing data as necessary. Finally, 5000.02 states that the use of Operational Test and Evaluation activities starts at Milestone C, where focus shifts towards system validation. The Operational Test and Evaluation Activity (OTA) plans and executes testing of production-level test assets in operational conditions to ascertain system performance and behavior.

Directive 5000.02 does not provide any guidance on test and evaluation activities past the Operational Test and Evaluation phase.

a. Systems Engineering (Enclosure 3)

Enclosure 3 of Directive 5000.02 establishes policies regarding the utilization of the systems engineering thinking and process to defense acquisition. Based on Enclosure 3, the Program Manager cooperates with the Lead (Chief) Systems Engineer to utilize the systems engineering for the entirety of the program life cycle.

Enclosure 3 provides more detailed guidance on the development of the Systems Engineering Plan (SEP) and the establishment technical risk management tools and associated metrics. Test and evaluation activities are involved in trade-off analyses, the definition of key performance parameters (KPP), and technical performance measures (TPM). Enclosure 3 directs programs to increase modeling and simulation integration in the acquisition process and to plan for reliability and maintainability (RAM) further in the system life cycle.

b. Developmental Test and Evaluation (Enclosure 4)

Enclosure 4 of Directive 5000.02 states that Developmental Test and Evaluation activities aid in risk identification, management and mitigation planning. It provides verification of requirements and specifications and characterizes system performance

prior to operational level tests. Test and evaluation results inform the decision-making progress and aids in the tracking of progress and technical maturity, ensuring technical maturity prior to milestone reviews

A key statement of Enclosure 4 is that Developmental Test and Evaluation planning requires integration from the earliest stages of the systems engineering cycle. This integration is particularly important during requirements definition as it ensures that requirements comply with the attributes defined in Chapter II. Enclosure 4 also reinforces the understating that early involvement of test and evaluation will prevent increased costs to repair deficiencies late in the system production cycle.

Another key mandate of Enclosure 4 is the need to integrate test and evaluation activities to reduce test activity redundancy and costs. This integration requires collaboration and cooperation by activities and personnel with the common goal of a rigorous and thorough test and evaluation program.

Enclosure 4 establishes that the Program Manager will designate a Chief Developmental Tester, as early as practicable and in accordance with Title 10 §139b, to support the Program Manager and Lead Systems Engineer. The chief developmental tester then forms and chairs the Test and Evaluation Working-Level Integrated Product Team (T&E WIPT). The key roles of the chief developmental tester are defined in Title 10 §139 and detailed in earlier in Chapter III. The role of the Test and Evaluation Working-Level Integrated Product Team is to aid in the development, execution and tracking of the test and evaluation plan, including the generation to the Test and Evaluation Master Plan.

c. Operational Test and Evaluation (Enclosure 5)

Enclosure 5 of Directive 5000.02 reinforces the statement that the test and evaluation activities are key for the generation of knowledge necessary to aid decision-makers to manage the acquisition process, reduce risk and monitor system performance. It also reinforces the guidance for program managers to integrate test and evaluation pre-Milestone A to develop a test program that spans the entirety of the program life cycle.

Enclosure 5 also establishes the following guidance:

- The Program Manager will form a Test and Evaluation Working-Level Integrated Product Team to develop and track the test and evaluation program, including the creation of the Test and Evaluation Master Plan. There is no clarification if this is the same Integrated Product Team formed by the Chief Development Tester in Enclosure 4.

- Design of experiments will be implemented during the test and evaluation program to establish the test methodology, specifications, parameters and operational conditions that provide complete coverage of the system evaluation effort.

- Test and Evaluation will be used to provide data to enable Reliability and Maintainability (RAM) and risk management planning and activities.

- Test and Evaluation will be used to address Cybersecurity vulnerabilities and interoperability.

- Test and Evaluation infrastructure, tools and resources will have documented strategies to ensure they are verified, validated or accredited, as necessary. The Program Manager and Test and Evaluation Working-Level Integrated Product Team are responsible for ensuring the accreditation status of test activities.

- Test plans should include details on test execution, required test order and data collection test-points.

DOD 5000.02 provides greater detail on the roles defined in Title 10 and provides further attributes for the systems test architect, particularly in the areas of process planning and integration.

4. *Defense Acquisition Guidebook*

The *Defense Acquisition Guidebook* is a complementary source of guidance to the policies in directives 5000.01 and 5000.02; it serves as a reference during the acquisition process. The contents in the guidebook are not mandatory expectation for program managers and other acquisition professionals, but its contents are regarded as a collection of best practices.

Chapter 9 of the *Defense Acquisition Guidebook* is devoted entirely to test and evaluation in the acquisition process. Most of the content of this chapter is covered in the

preceding sections, some of the additional guidance and best practices exposed in this chapter include:

- The Program Manager will base developmental decisions based on planned and documented test events, to the greatest extent practicable.
- Test results should be repeatable or statistically defensible, especially if they are for planning purposes.
- Design of Experiments should be used to optimize test scheduling, test data collection strategy and test asset utilization.
- Test and evaluation activities should be integrated from the earliest staged of the acquisition process; this requires early involvement of the test and evaluation workforce to seek their input in the early requirement and capability definition stages.
- Requirements, Measures of Effectiveness, Measures of Performance and Technical Performance Measures should be allocated to test events with associated test plans in a matrix format that clearly maps their relationship.

5. *Test and Evaluation Management Guide*

The *Test and Evaluation Management Guide* (TEMG) is a document prepared by the Defense Acquisition University (DAU) in support of their coursework, but also intended as a desk reference for program managers and test and evaluation workforce members. It is written in support of both Department of Defense and industry test and evaluation efforts. Like the *Defense Acquisition Guidebook*, the contents of the guide are non-mandatory; rather, this guide is a collection of educational material and best practices.

The *Test and Evaluation Management Guide* reinforces the belief in the value of test and evaluation as a tool that aids the acquisition process throughout the program life cycle. Most of the content of the *Test and Evaluation Management Guide* has already been covered previously, some of the guidance and best practices detailed in this guide include:

- Program decisions are based on test and evaluation event data and documentation such as the Test and Evaluation Master Plan, test reports, test data, simulation data and analyses.

- A methodology or process should be established for identifying, tracking and reporting system deficiencies found by test and evaluation activities.

- The Systems Engineering Plan, Test and Evaluation Master plan and individual test and evaluation plans should mutually consistent.

- The Test and Evaluation Master Plan should take into consideration the design of the entirety of the test and evaluation program (e.g., resources, personnel, requirements, review, roles, responsibilities).

- The test and evaluation involvement should follow a clearly defined model similar to that in Figure 5.

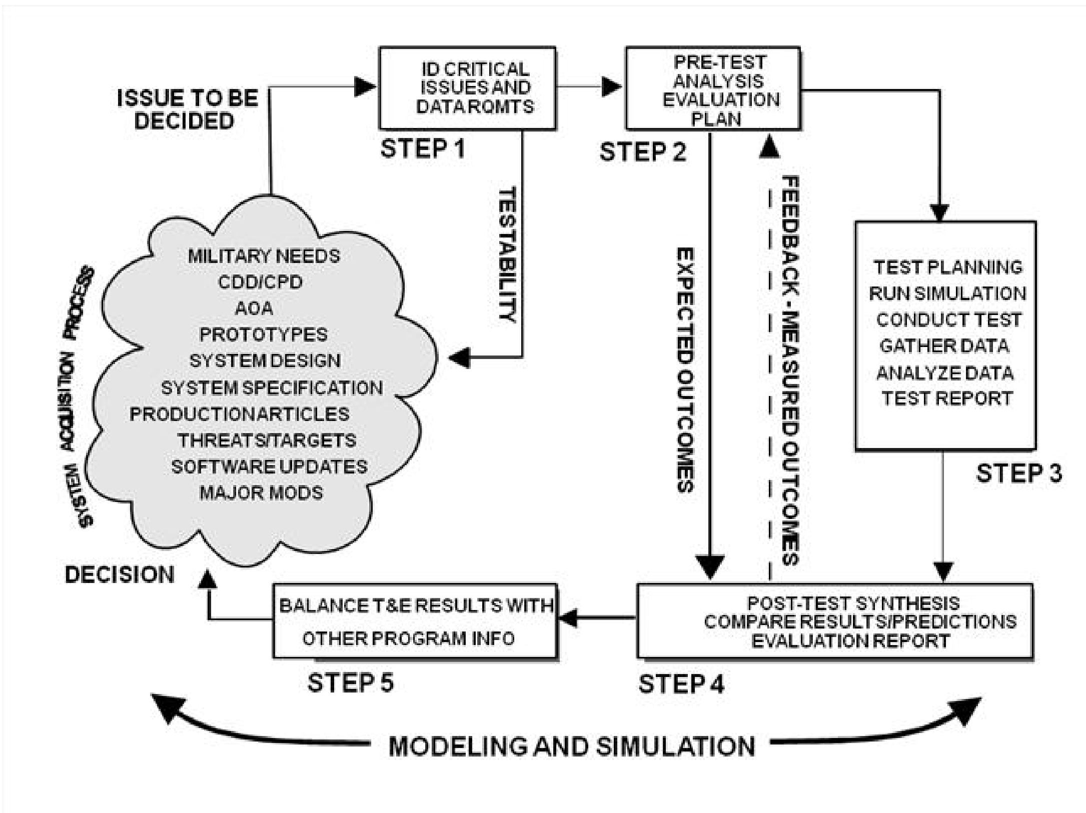

Figure 5. Five-Step Test and Evaluation Process. Source: United States Department of Defense (2012)

- Acquisition programs should have an element or party responsible for the entirety of the test and evaluation program across all phases. In the Test and Evaluation Management Guide this role is attributed to the Chief Developmental Tester; however, the Chief Developmental tester does no participate the Operational Test and Evaluation activities.

- The timing of the progression from developmental testing operational testing should be well though-out to ensure that all capabilities have been fully developed.

- Data sets should be optimized to ensure their usability by the greatest number of tools, methods and processes as possible.

- Identify, as early as practicable, the test resources and infrastructure necessary to execute the test and evaluation program.

- Manage test articles or assets, users, personnel, facilities, ranges and instrumentation to ensure total system validation.

- Utilize modeling and simulation in coordination with test and evaluation to produce the most representative models that inform decision-makers.

- Utilize modeling and simulation to prepare dynamic test plans that can be tailored as necessary based on test results.

The Test and Evaluation Management guide provides a scattered but extensive list of responsibilities for the chief developmental tester (or the equivalent person) that includes:

- Plan, direct and oversee Developmental Test and Evaluation activities.

- Review test approaches and test plans to ensure test adequacy and gauge the correct level and rigor of testing.

- Document decisions made based of test data.

- Utilize and collaborate with test and evaluation subject-matter experts to accomplish test and evaluation activities and ensure the use of best practices.

- Manage test program funding, estimates and statements of work.

- Set up a data collection infrastructure.

- Identify and coordinate test resources, assets, support and infrastructure.

- Coordinate Test Readiness Reviews.

- Participate in technical reviews.

- Support in the creation of test and evaluation portions of proposal requests.

- Coordinate the sharing of test plans, data and related documentation across departments and agencies, as well as the format for those documents.
- Coordinate the test schedule.
- Communicate frequently to the Program Manager the status of test results, risk items, technical performance measures and other parameters of interest for program decisions.
- Act as advocate for test and evaluation activity funding.
- Review specifications for adequacy and testability.
- Monitor and review test events and results.
- Ensure integration of Operational Test and Evaluation activities across the acquisition life cycle.

The definition of the role of chief developmental tester is a great leap toward the integration of test and evaluation is the systems acquisition process. It is a model to emulate despite the lack of ownership of the operational test phases.

B. INDUSTRY GUIDANCE ON TEST AND EVALUATION

Individual companies in industry do not commonly release detailed information on their guiding policies and processes. More commonly, companies or their members partner and collaborate to publish guidance to serve as standard for other companies to follow. Such is the case for the International Council on Systems Engineering (INCOSE), a collaborative body founded by systems engineering professionals to collect and disseminate knowledge about systems engineering. Their main goal is to “develop and disseminate the interdisciplinary principles and practices that enable the realization of successful systems” (INCOSE 2016a).

INCOSE achieves this by collecting and sharing information about systems engineering, promoting collaboration and encouraging investment in systems engineering by governments and industry. Its publications serve as guidance, and the organization is a forum for members to engage and help develop best practices in systems engineering. The collection of knowledge developed by INCOSE is a major source of reference material for the Systems Engineering Body of Knowledge (SEBoK) website. Among the

publications it develops are two guides related to the use of test and evaluation. As a voluntary organization, the guidance set forth by INCOSE is non-mandatory. It is up to the individual organizations to adopt and implement that guidance.

1. INCOSE-TP-2003-020-01 – *Technical Measurement*

This guide defines and describes how to select and use Measures of Effectiveness (MOE), Measures of Performance (MOP), Technical Performance Measures (TPM) and Key Performance Parameters (KPP) for successful project management. This guide uses the term "measurement" throughout to describe the set of activities that encompass test and evaluation. Overall, the Department of Defense guidance is clearer about the relationship between the technical and managerial aspects of systems engineering.

TP-2003-020-01 describes measurement as having the following benefits:

- defines and tracks a technological solution
- identifies and manages risk
- tracks technological maturity
- improves decision-making
- increases project likelihood of success
- informs the trade-off analyses

This guide defines the four levels of measurements hierarchy (Figure 6) as being an interdependent set of measurements, used throughout the systems engineering process; these are the same measurements defined and used by the Department of Defense.

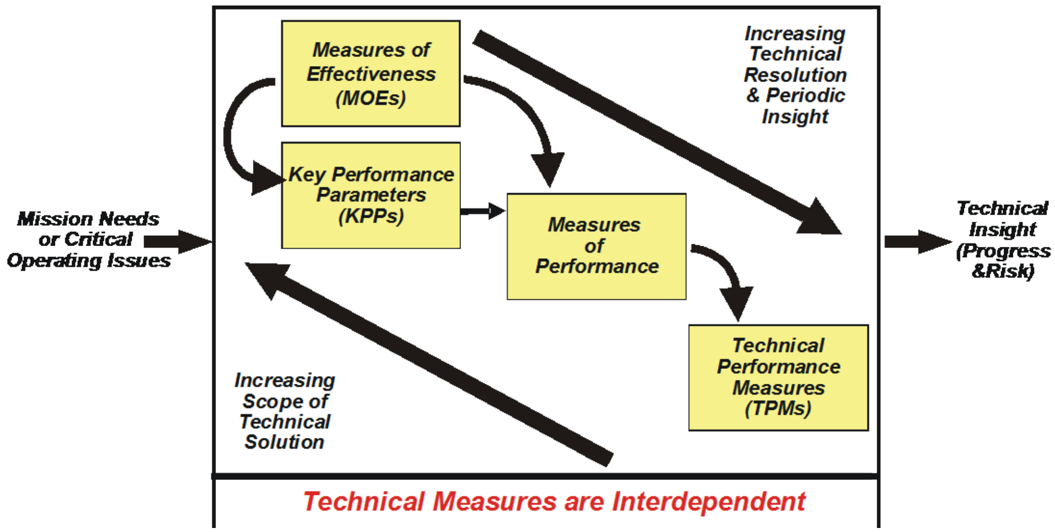

Figure 6. Relationship of the Technical Measures. Source: International Council on Systems Engineering (2003)

Aside from defining the hierarchy of measurements, the guide assigns attributes to measurements such as thresholds and variances. These attributes become useful in gauging the usefulness of each measurement to the systems engineering process. These attributes can be used to define and track the current state and history of a measurement.

The guide provides three key concepts related to the measurement process, and a basic model for the involvement of the measurement process in the systems engineering process.

- The measurement process must be tailored to the specific needs of the project at hand.
- The selected measurements, their objective and importance must be clear to all parties.
- The measurement process must be utilized in order to provide value to the project.

While the guide discusses at length the various measurements and their use, the guide lacks any guidance or best practices on the responsibilities for planning, organizing or conducting the measurement activities detailed. The only guidance related to the management of the test and evaluation program is the usage or integrated product teams or project team to support the measurement process.

2. INCOSE-TP-2010-005-02 – *Systems Engineering Measurement Primer*

This guide is a more generic introduction into the utilization and involvement of measurement activities in the systems engineering process. Despite this more generic approach, it does provide some valuable guidance on the relationship between these two processes including:

- effective use of measurement
- issues to avoid
- selection of measurements
- benefits of measurement

Although this document does not describe any roles specific to measurement, a valuable contribution of this document is in the form of "tips," including:

- build trust between the program management and measurement activities. Management should view the measurement process as an asset, rather than an adversary to project funding and success
- select only those measurements that provide value to the project (e.g., performance, risk)
- assign an owner to the measurement process
- assign a measurement process to every measurement
- trace every measure to a requirement or issue

C. CHAPTER SUMMARY

This chapter presented description and discussion of the DOD and documented industry policy and guidance on test and evaluation. Figure 7 depicts the content of this chapter and indicates the relationships between the team members and the different test and evaluation facets. From this chapter, it is evident that DOD makes great strides to tackle both the managerial and technical aspects of test and evaluation within the systems acquisition process, while industry is more focused on the technical aspects. DOD does define a specific role to aid in the development and management of a test program, but the role is defined in the context of large ACAT I/II programs of record and limited to

developmental testing. This prompts the question that forms the basis for this research, what happens for a small development task?

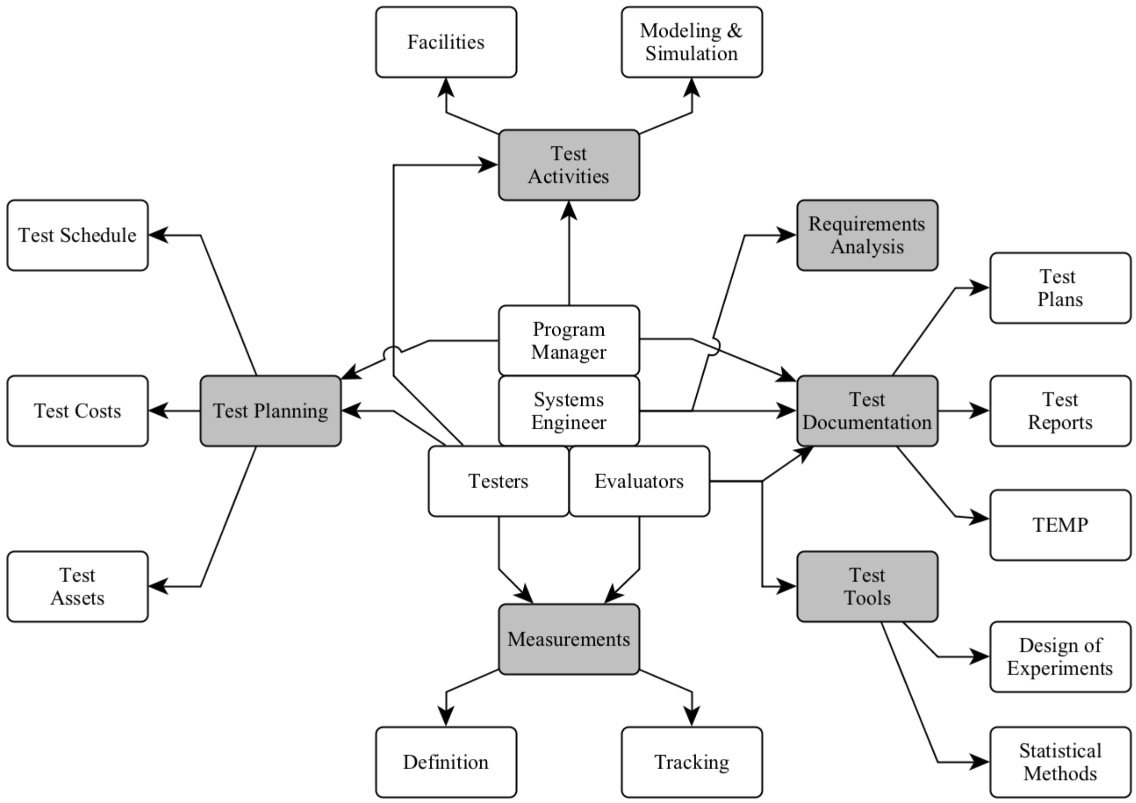

Figure 7. System Engineering Test and Evaluation Responsibilities

The next chapter begins to address that question, what roles and responsibilities must the systems test architect embody in order to enable the early integration of test and evaluation in systems engineering. Next chapter takes the scattered set of responsibilities and molds a model for the systems test architect and their fit in the systems engineering process.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. THE FIT OF THE SYSTEMS TEST ARCHITECT

This chapter is an exposition and discussion of the potential fit of the systems test architect in the systems engineering process. That is, what are the roles responsibilities and interactions of the systems test architect? Title 10 and DOD 5000 have an adequate description for a similar role in the form of the chief developmental tester combined with the test and evaluation working-level integrated product team (T&E WIPT), but industry-wide or in systems engineering textbooks there does not appear to be any such documented description.

A. ROLES OF THE SYSTEMS TEST ARCHITECT

The main role of the systems test architect is that of the "systems thinker" (Brewer, Emmert, and Guise, 2012) for test and evaluation. The systems test architect takes a holistic view of test and evaluation, ensuring that test and evaluation is integrated throughout the system life cycle. Regarding the future of test and evaluation, Bodmer (2003) recommends that test and evaluation must become "more agile and more embedded in the process of acquisition," to this end, the goal of the systems test architect is to embed him or herself within the systems engineering process as the embodiment of the test and evaluation program.

Although many of these roles are indicative of the chief developmental tester and test and evaluation working-level integrated product team, those roles are defined in a much different scope, meant for the acquisition or large, congressionally approved programs of record. For generic tasks that utilize systems engineering there is only a small inroads effort to define this type of role.

An important distinction is that the systems test architect is not the tester, but a coordinator, enabler, advocate and liaison between the program manager, system engineer, developers and the test and evaluation workforce. Figure 8 shows the potential placement of the systems test architect in the program structure. In the current paradigm, the management and execution of test and evaluation program is dispersed across the program structure, with no leader other than the program manager and no domain expert

unless a member of the systems engineering team engages an external party. The systems test architect gathers these roles into a single domain-expert-driven role.

One role that is not in this discussion is that of manager, and that is because the systems test architect does not manage personnel. The systems test architect engages others constantly, and has authority over the test strategy planning and execution, but that authority is only over the process.

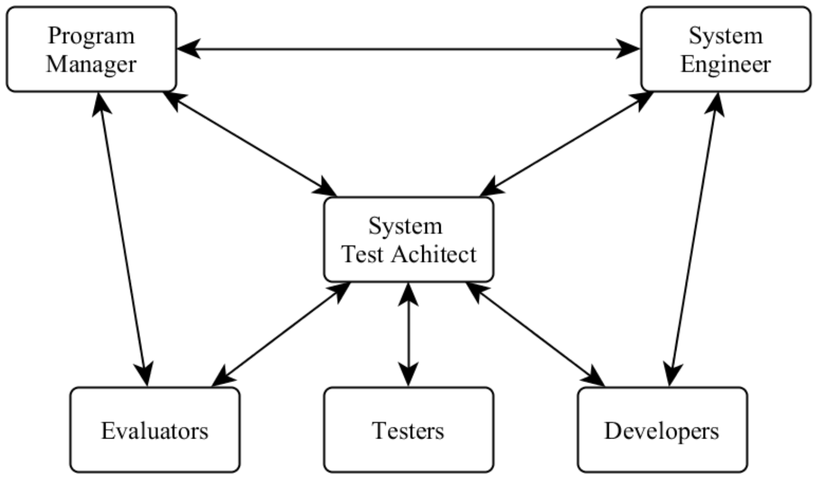

Figure 8. Systems Test Architect Influence Model

1. Advocate

The systems test architect is the advocate for the test and evaluation program, the voice of the testers in the systems engineering process. Much like the chief developmental tested in Chapter III, the systems test architect ensures that throughout the system development effort, test and evaluation activities engage in and influence the process. A rigorous and thorough test and evaluation program is the backbone an event-driven systems engineering schedule. Program managers and the systems engineering should strive for such an event-drive schedule, although achieving a purely event-drive schedule is a utopian goal. Without proper representation, program managers often skip the test and evaluation activities and the knowledge required to inform decisionmakers is not available.

The systems test architect is the embodiment of the test and evaluation master plan. The systems test architect seeks out the planners, testers, operators and evaluators to build the test and evaluation strategy, influence planning, schedule, metrics and methods. The systems test architect works with programs managers and the systems engineering team to utilize the test and evaluation workforce capabilities.

2. Steward

The systems test architect is the steward of the test and evaluation program. The systems test architect provides expertise, information and clarity regarding test and evaluation needs, events and results during program, technical and readiness reviews.

As Barret (2009, 38) states,

> There is no standardized process for determining who (by function or level of authority) should participate at each review from the T&E workforce. As a result of this lack in guidance, the selection of T&E personnel to participate in SETR events tends to be ad hoc. This results in inconsistent representation of the T&E workforce both across programs and across SETR events within the same program.

The systems test architect fills this gap in representation at reviews and creates consistency in test and evaluation involvement in the system development.

The systems test architect works to keep the test and evaluation program off the “chopping-block,” and ensures that program managers do not compromise the program in pursuit of shortened schedules. In a study of test and evaluation best practices the United States General Accounting Office (2000, 50) details the failure of the DarkStar unmanned aerial vehicle. In this example, the program managers did not emphasize the test and evaluation needs and ignored the feedback from testers regarding the design, all because they were schedule-driven. The DarkStar program was eventually cancelled at the expense of United States taxpayers.

The systems test architect works to embed the test and evaluation strategy into the entire life cycle. As the systems thinker for test and evaluation, the systems test architect is working to: develop plans, schedule tests and establish metrics and methods used to

analyze requirements and validate system functions. This systems thinking also includes planning for production, maintenance and upgrade test requirements.

3. Advisor

The systems test architect is the advisor for the program manager and systems engineer on test and evaluation matters. This is similar to what the chief developmental tester and the test and evaluation working-level integrated product team accomplish, as described in Chapter III. The systems test architect can identify risks related to testing, such as test resource availability, conflicts in schedule, and test costs.

The systems test architect evaluates requirements for test issues, such as gauging technology levels and methodologies to determine if a requirement is feasible and testable. If the technology or methodology to perform a test does not exist, the systems test architect can engage the test and evaluation workforce to develop that technology or recommend alternate requirements that are testable. This exercise in requirements analysis allows the systems test architect to identify possible investments in test infrastructure to the program manager.

The systems test architect communicates the state of critical performance parameters to the program manager and the systems engineer and advises on the meaning of the state of these parameters. These critical parameters are usually the ones decisionmakers utilize to gauge the system maturity, having the systems test architect provide context and meaning ensures that decisions are knowledge-based. To that end, the systems test architect also provides feedback to the program manager and evaluators on test results.

The systems test architect works to ground the expectations of the program manager with respect to the value, cost, capability and impact of test and evaluation. The systems test architect serves as constant reminder to program managers that "The test organization, test management and test engineers are essential elements of the program management team from program initiation" (Science Applications International Corporation 2002, 17). It is also the authors' experience that program managers and

system developers fail to involve the test and evaluation workforce, a practice that often leads to confusion, last-minute churn and rework at test time.

The systems test architect works to maintain the validity of test methods throughout the evolution in requirements; that is, as requirements change to arrive at their final values, the test methodology remains as constant as possible. Only the metric, not the method, should change. During requirements reviews and updates, the systems test architect works with the program manager and systems engineer to ensure that requirements changes remain within the same method. For example, if the requirement is for some maximum stress, then the modified requirement should not be for maximum deflection, as the methods to test these two items are different.

During the creation of the test and evaluation master plan (TEMP) and systems engineering plan (SEP), the systems test architect works with the systems engineer to identify system design-maturity snapshots. During these snapshots, the system state is fixed based on some individual maturity metrics, at which the system attributes are tested and the results fed back to developers. This allows the systems test architect and systems engineer to correct any deficiencies earlier in the development cycle.

4. Domain Expert

The systems test architect is the domain expert and subject-matter expert on test and evaluation. This requires that the systems test architect be an experienced test and evaluation workforce member, knowledgeable of test facilities within and without their organization. The systems test architect requires familiarity with the capabilities and limitations of each of these facilities and how they relate to the system at hand.

The systems test architect must be knowledgeable of test methods, and how to establish and track metrics. The systems test architect must know how and when to integrate modeling and simulation applications to augment test capabilities and reduce the number of tests.

As the domain expert, the systems test architect works with systems developers to influence the system design for testability. That is, the systems test architect will work to

have test points, ports, and other features that aid in testing be built-in to the design from the earliest stages. The systems test architect will work with testers and evaluators to generate the required test scenarios from these early stages.

When the system is tested, the systems test architect converts test data into meaningful information. Then evaluates the data for unexpected results and provides that feedback to developers. In this influence over design, the systems test architect is the counterpart to the systems architect.

Guided by their experience, the systems test architect maintains a consistent, rigorous and quality-driven approach to the test architecture.

5. Communicator

The systems test architect is an enabler of communication; working to maintain a constant cycle of communication up and down the program hierarchy, from program manager and systems engineer to developers, testers and evaluators. The systems test architect, as proxy stakeholder for test and evaluation, is able to maintain an objective relationship with the program manager in lieu of the often contentious relationship between the two. The systems test architect works to promote a culture of cooperation throughout the program and across programs as the systems test architect cooperates with other systems test architects to coordinate and share resources, test models and knowledge.

The essence of communication is systems engineering is the eliciting, sharing and coordination of knowledge. The systems test architect works to gather that knowledge, which is dispersed amongst the program members. The systems test architect asks the questions that help shape the test architecture. Among those questions the systems test architect asks are:

- How will the requirement, metric or risk be tested?
- When and where will the test take place?
- Who will perform the test?
- What resources, assets and personnel are needed to perform the test?

- What is the target value?
- What is the expected tolerance?
- How can modeling and simulation be used?
- Can tests be combined or reduced? How?
- How does the result affect the process?

6. Stakeholder

The systems test architect is the proxy stakeholder for test and evaluation. Ultimately, the system needs to be tested; it will have to go from concept to reality. In order to make that transition, testability has to be designed-in, it has to become part of the systems engineering process. As a stakeholder, the systems test architect holds influence; which is now diluted across other members of the systems engineering team. This dilution of influence makes the voice of the test and evaluation workforce have less sway.

B. RESPONSIBILITIES OF THE SYSTEMS TEST ARCHITECT

In order to fulfill the roles attributed previously, the systems test architect takes ownership of the test and evaluation program. The program manager confers this responsibility to the systems test architect, much like the chief developmental tester in Title 10. The systems test architect then has the responsibility for the development and execution of the test and evaluation strategy and plan. The test architecture is the tapestry formed by the test and evaluation strategy and plan and these term are used interchangeably in this research.

The test architecture is the tapestry or framework of test activities that will guide the development of the system from conception to delivery. Figure 9 provides a map of the scope of the responsibilities of the systems test architect.

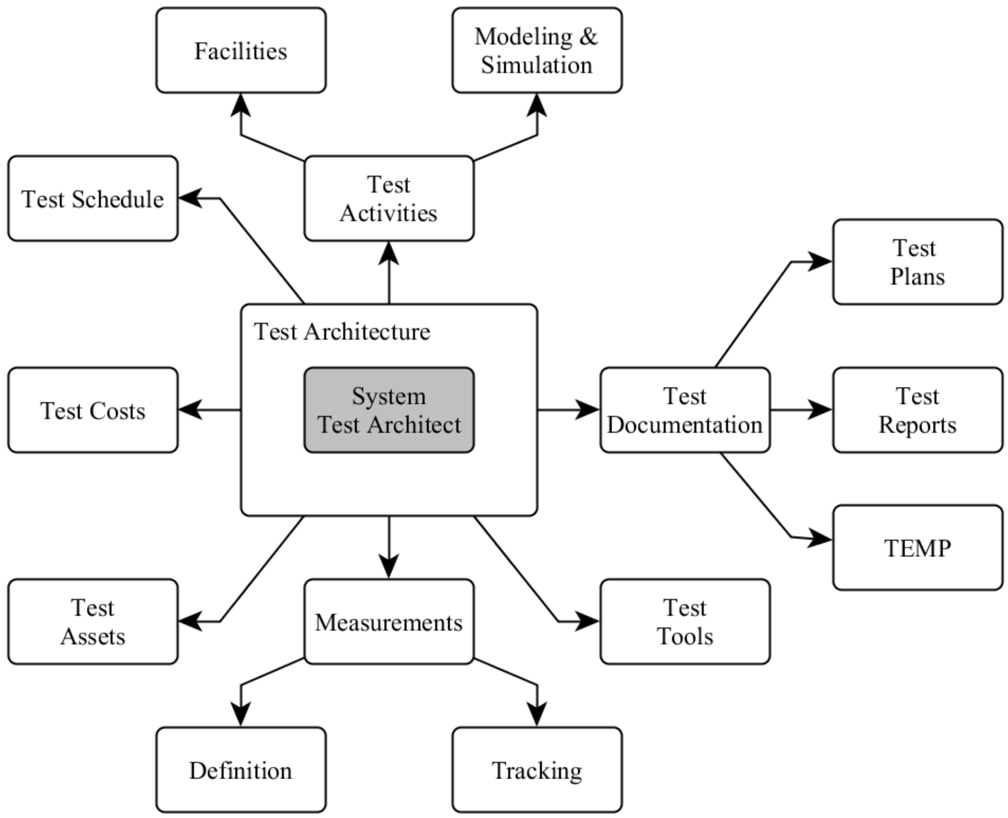

Figure 9. Systems Test Architect Responsibility Map

1. Development of the Test and Evaluation Strategy

The systems test architect develops the test and evaluation strategy; this strategy should be objective, repeatable, statistically defensible, disciplined, and test-event driven. While in Chapter III this is identified as function of the test and evaluation working-level integrated product team, the systems test architect fulfils this function in the smaller scope of systems engineering The impact of a "good" test and evaluation approach is not only evident in the short-term, as requirements are met; but in the long-term as the life-cycle costs are reduced. A rigorous and thorough test and evaluation strategy should result in earlier identification of system deficiencies and defects.

To create the test strategy, the systems test architect utilizes the requirement and architectural decompositions to identify the needed test requirements. This includes identifying the test personnel, assets, infrastructure and modeling capabilities that will be

used to test, as well as developing the criteria by which the data will be evaluated. As previously described, test facilities require some degree of certification, the systems test architect reviews and coordinates certification of test activities and modeling tools.

The systems test architect engages the test and evaluation activities to create estimates for the cost of test. This includes internal testing conducted by the organization, and external testing performed by some contractor. Estimates include not just the monetary cost of the test but also the resources, time and personnel required. The systems test architect guides the testers in the creation of the test plans, ensuring a consistent level of quality to the test strategy. Using these estimates, the systems test architect can create a budget for the execution of the test and evaluation strategy, which is given to the program manager to approve and fund. It is then the responsibility of the systems test architect to manage and maintain that budget.

The systems test architect is in charge of the creation of the test schedule, this is an event-driven schedule showing the sequential relationships between test events. The goal of this schedule is to allow the systems test architect to determine which tests may be run in parallel or combined and determine where risks to the test schedule lie. During the creation of the test schedule, the systems test architect identifies areas where test events can be combined or reduced through cooperation with other related tasks or through the use of modeling and simulation. The systems test architect also looks for areas in which to bridge the gap between developmental and operational testing (verification and validation).

Design of experiments is a key tool for the systems test architect. Using design of experiments, the systems test architect focuses the scope of testing. Using design of experiments, the systems test architect ensures the greatest number of requirements is tested with the least number of tests. The goal of each test event is to provide the greatest amount of knowledge, tests are selected because they fill in the blanks in the tapestry of the test architecture. Another key tool related to design of experiments is the use of statistical methods to ensure the right amount of testing. Statistical methods allow the systems test architect and evaluators to select the test of test conditions and scenarios that produce the best possible set of data.

Key components of the test strategy are the technical performance measures and critical performance measures, or key performance parameters. The systems test architect works with the program manager and the systems engineer to establish which of these parameters are technology maturity metrics. The systems test architect monitors and track this subset of measures to aid the decisionmakers to evaluate the state of the system. The systems test architect determines the data required to measure these metrics, establishes the methods and frequency of measurement and the evaluation criteria. The systems test architect, program manager and systems engineer then set the milestone entry and exit criteria that are dependent on these measurements.

As the systems thinker for test and evaluation, the systems test architect is planning the test and evaluation activities for the entire life cycle of the system. The systems test architect establishes the test requirements for each life-cycle stage, the method and metric used to evaluate the system. This includes not just the tests for the development, verification and validation of the system. It also includes testing related to production, maintenance, and updates. In the case of software, this would include some form of regression testing.

The systems test architect documents the test and evaluation strategy in the test and evaluation master plan. In the test and evaluation master plan the systems test architect includes a test-requirement traceability matrix or requirements verification matrix, showing the relationships from requirements to test events and scenarios. In this matrix, the systems test architect can allocate test events to requirements and risks, ensuring full coverage of the test strategy.

Within the test and evaluation master plan, the systems test architect creates the distinction on which tests encompass the verification and validation phases, and defines when they will take place based on entrance criteria. The systems test architect evaluates and documents the instrumentation requirements for each test to ensure the fullest range of requirements are evaluated on any given test event (Mosseau 2004, 72), and evaluates data requirements of the evaluators to ensure that measurements and information are compatible and usable across the greatest range of evaluation tools. The systems test

architect works with the evaluators to coordinate the data evaluation strategy, ensuring that common tools, methods and criteria are used uniformly across the program.

The systems test architect works with the program manager and the systems engineer to align the contents of the test and evaluation master plan and the systems engineering plan. These two documents are symbiotic and should reflect common goals, have a common language, agree on milestones, define a common set of measures (of effectiveness and performance), key parameters and technological maturity measures.

2. Execute the Test and Evaluation Strategy

The systems test architect executes the test and evaluation strategy. This activity requires constant communication and interaction of the systems test architect with the system engineering team, testers and evaluators. The systems test architect is fully aware that the test architecture and the test and evaluation master plan are mutable and is adapted as necessary based on feedback from the process.

During the execution of the test strategy the systems test architect continually monitors and evaluates the technology maturity metrics. Based on the state of these metrics the systems test architect makes recommendations on whether the system is ready for the next development phase or if additional development is necessary. The systems test architect also tracks the state of critical performance parameters unrelated to maturity, and will re-evaluate the strategy as necessary to lessen any impacts to the test schedule.

The systems test architect reviews the test plans generated by the testers and participates in test readiness reviews. The systems test architect seeks to harmonize test plans to reduce redundancy and ensure the fit of the plans in the test strategy. The systems test architect reviews the test plans for completeness and quality and ensures that tests are designed to provide knowledge and not designed to pass.

The systems test architect coordinates the test events in the schedule. The systems test architect prioritizes test events as necessary to accommodate test asset availability, time, cost, order of execution or some pre-requisite of knowledge. This coordination

occurs across the organizations' and contractors' test activities. The systems test architect also coordinates the required test assets, ensuring that the right asset is available for each test. The systems test architect will then act as witness for test events; this first-hand experience provides better understanding of the sequence of events during the test and provides the context the systems test architect needs to analyze results and formulate recommendations.

The systems test architect reviews, recommends approval of the test reports generated by the testers, and performs the initial evaluation of the test results, the approval of the test plans remain the responsibility of the program manager. The systems test architect will work to identify issues and ensure the completeness of test approach; by completeness it is meant that every aspect of the test plan is addressed as part of the plan. Based on the evaluated test results, the systems test architect will then make recommendations to the program managers, systems engineers and developers on how the test result relates to the parameter of interest.

Throughout the execution of the test and evaluation strategy, the systems test architect maintains the test and evaluation master plan, and performs updates based on results of test events and evolution of the program. This includes maintenance of the test-requirement matrix and ensuring that by end of the verification and validation phases all requirements are thoroughly tested. Finally, the systems test architect documents the lessons learned, recommendations and associated rationale so that future endeavors benefit from the successes and failures of the program execution.

C. INTERACTIONS OF THE SYSTEMS TEST ARCHITECT

The systems test architect is, as seen from the previous discussions, an enabler. Principally, and according to design, the systems test architect enables earlier and more thorough integration of the test and evaluation efforts with the systems engineering process. That is, the systems test architect enables the "shift left" mentality that DOD desires. This requires open and constant communication up and down the development chain, from program manager to testers. The systems test architect takes on the interactions that the program manager and system engineer would incur, should they have

to domain knowledge required. It is this domain knowledge and relative independence from the program manager that provides the greatest benefit to these interactions.

1. Program Manager

The program managers' focus is the overall execution of the development effort. The systems test architect, as advisor to the program manager, will make recommendations on system maturity and state based on test results. The program manager has the ultimate responsibility for acting upon this advice. The systems test architect regularly updates the program manager on system maturity level and advices on the readiness of the system to continue from one phase to another. The systems test architect and the program manager work together to coordinate the entrance and exit criteria for each of these milestones.

After review, the systems test architect recommends the approval of test plans and reports. Giving the program manager an assessment of these documents and providing the domain-expert interpretation of the results of the test.

The systems test architect works with the program manager to ensure that adequate resources, personnel, assets and funds are available to execute the test strategy, and make contingencies for budget shortfalls, overruns of schedule compression. The systems test architect will focus on keeping the test program an integral part of the systems engineering effort without becoming a financial burden to the program manager.

2. Systems Engineer and Systems Architect

The systems engineer and the systems test architect are team, they work in tandem with the systems architect to perform the requirements definition and architectural decompositions. The systems test architect will make recommendations on systems requirements based on testability of requirements and test results. The systems test architect will also make recommendations and provide feedback to the systems architect on physical architecture arrangements. If a particular arrangement is too difficult to understand and test, the systems test architect may provide means to simplify the architecture and interfaces to improve testability. The physical architecture may also be

affected by viable test methods; the systems test architect would be able to identify areas where an architecture is simply infeasible.

The systems test architect works with the systems engineer to identify deficiencies and issues of compatibility or interoperability. Together, the systems engineer and systems test architect break down requirements into measures of effectiveness, measures of performance, key performance parameters, technical performance measures and together with the program manager they identify the system maturity metrics. The systems engineer and systems test architect align the systems engineering plan and the test and evaluation master plan, ensuring the common goals of the two documents.

3. Testers and Operators

The testers are the core of the test and evaluation workforce, they are the ones that are familiar the operation of the test facilities, they have the first-hard experience of what is feasible. Testers know the limits of their facilities, and the requirements for operation. The operators are the intended users of the system, sometimes the testers will act as operators during developmental testing, but most times the operators are a distinct group with invaluable insight into the system design.

The systems test architect works with testers to develop the individual test plans. Testers will create test plans based on test scenarios that are developed with the systems test architect, then the test plan is reviewed by the systems test architect and approved by the program manager. Together, the testers, operators and systems test architect coordinate the data and instrumentation requirements for the test and evaluation strategy.

The systems test architect engages the testers and operators to request their input on requirements testability, feasibility, test costs and scenario development. The systems test architect will also involve the testers and operators should a particular requirement require some initial evaluation through testing or modeling and simulation

4. Developers

The systems test architect works with developers to drive the system design to incorporate test point integration and overall design for test. After a test event, the systems test architect provides feedback on system design based on the test results.

The systems test architect influences the developers to influence their design decisions for improved reliability, availability and maintainability (RAM), improved system safety, life cycle and interoperability

5. Evaluators

The systems test architect works with the evaluators to create a uniform evaluation strategy. This includes agreeing on data requirements, evaluation tools and statistical methods, analysis techniques and reporting format. While the evaluators will work to glean as much knowledge from the results of a test, the systems test architect provides meaningful context and clarification of test results. The systems test architect is the initial lens through which results are interpreted, while the detailed analysis of the data remains the responsibility of the evaluators.

D. CHAPTER SUMMARY

This chapter presented the model for the role of systems test architect, broken down into roles, responsibilities and interactions. The systems test architect is the proxy stakeholder for the test and evaluation processes, the embodiment of the test and evaluation master plan, advocate for test and advisor on all things related to test and evaluation. The systems test architect communicates across the organization to create and execute the test and evaluation strategy and to integrate test and evaluation into the systems engineering process as early and as often as possible.

The next chapter presents a discussion on the expected value of the role of systems test architect to the system engineering process. Should the program manager establish such a role in the program structure, encompassing the model presented in this chapter; what value could be reasonably expected?

THIS PAGE INTENTIONALLY LEFT BLANK

V. THE VALUE OF THE SYSTEMS TEST ARCHITECT

This chapter presents a discussion on the potential value of the systems test architect to the systems engineering process. Program managers are not likely to appoint another member to the program structure without value added. Demonstrating this value added is an continuing struggle in systems engineering. By itself, integration of test and evaluation into the systems engineering process can be very valuable, as discussed in Chapter II. In this chapter, the discussion focuses on what value the systems test architect can potentially provide, in the execution of the model presented in Chapter IV.

A. AREAS OF VALUE ADDED

1. Process Improvement

The systems test architect provides the systems engineer many of the benefits that the chief developmental test and test and evaluation working-level integrated product team combined confer to the acquisition process in DOD 5000. The difference, as stated in Chapter IV, is that these roles are defined in the scope of ACAT I/II programs of record and are tailored as such. One can view the roles of the systems test architect as a meta-model for these two roles, since in they are a decomposition of the roles and responsibilities of the systems test architect in the framework of the DOD acquisition process.

The systems test architect addresses and encompasses many of the best practices detailed in (Science Applications International Corporation 2002, 2–4), these are the value-added attributes of early involvement of test and evaluation involvement in systems engineering. The systems test architect:

- provides senior personnel a link to test and evaluation knowledge and understanding
- provides early and consistent involvement of test and evaluation in program planning
- stabilizes the relationship between program management and testers
- determines investment required for test resources and capabilities

- estimates the cost of testing at program start
- ensures the test and evaluation workforce and developers are in synch with the need of one and other
- integrates program planning and test and evaluation activities
- integrates the use of metrics to enable knowledge-based decisions

Someone in the systems engineering process must be mindful of the need to test: how it will be done, by whom, where, when and how the results of that testing feeds back to and influence the process. The systems test architect provides that mindfulness, embraces it and makes it a reality.

The systems test architect ensures that by the end of the verification and validation phases, all requirements have been addressed and the test process is documented along with the results, lessons learned and any necessary rationale. This allows future efforts to benefit from the successes and failures of a particular effort, and improve their implementation of the system engineering process. The systems test architect provides a continuity of support to the system engineering process that is difficult to reproduce with the integrated product team approach of DOD 5000. The systems test architect is continually involved with the technical and interpersonal portion of the process, integrated product teams meet only periodically and members may rotate, taking their knowledge with them.

The systems test architect helps formalize the involvement of test and evaluation in the systems engineering process much like the systems architect addresses architecture definition and decomposition. The systems test architect is the counterpart of the system architect for test and evaluation. The systems test architect advocates for the involvement of test and evaluation based on the following tenets: no test is a waste of resources if knowledge is gained, a test is not a pass/fail gauge of system maturity, test events are discovery events. Having a dedicated proxy stakeholder for test and evaluation in the systems engineering process is another way to improve the involvement of test and evaluation in system engineering and increase the probability of success. The systems test

architect helps the systems engineering process to become more event-driven by allowing more decisions to be based on knowledge rather than schedule.

The systems test architect alleviates the programmatic burden on the program manager by taking responsibility for the development and execution of the test strategy. The systems test architect also provides the program manager a constant avenue of communication to the test and evaluation activities. Much like the chief developmental tester in DOD 5000, inputs from the systems test architect "to the contract, engineering specifications, systems engineering efforts, budget, program schedule, etc., are essential if the PM is to manage T&E aspects of the program efficiently" (Defense Acquisition University 2012, 47).

In a study of test and evaluation best practices Science Applications International Corporation (2002, 9) details six traits, the combination of which drive the successful integration of test and evaluation in the systems acquisition process. The systems test architect can be seen as the embodiment of these traits of: providing stability, focus, consistency, commitment, domain-knowledge and objectivity to the test and evaluation aspects of the process. The systems test architect ensures a consistent, rigorous and knowledge-driven approach to the test strategy development and execution, instead of focusing on time schedules and success-focused tests. The systems test architect reviews test plans, oversees their execution and then reviews the data produced to ensure results are credible, repeatable and statistically defensible. The systems test architect optimizes the use of test and evaluation by using a holistic view, a systems view, involving: tools, instrumentation, facilities, ranges, modeling and simulation, strategies, assets and personnel.

The systems test architect "looks ahead," beyond the development of the system and into the operation and sustainment phase to influence design for future test needs resulting from reliability and maintainability activities.

2. Improved Requirements Definition

The systems test architect serves as a litmus test for requirements definition. The domain knowledge and resources of the systems test architect allows requirements to be

judged by their attributes of testability, clarity and consistency. If the systems test architect finds requirements that are untestable or infeasible, those requirements can be rewritten before decision-makers compromise the system development effort.

The systems test architect ensures the convergence of tests and requirements, by developing and monitoring the requirements verification matrix. The systems test architect focuses on ensuring that at the end of the development all requirements have been tested and the results documented.

3. Improved Communication

One of the most important benefits of the systems test architect is the open channel of communication created between the program manager and the test and evaluation workforce. Under the current paradigm this relationship is often contentious (United States General Accounting Office 2000, 9) and strained, based on the program managers view that test and evaluation results may lead to funding cuts, schedule delays and/or program termination, while testers often feel less than valued by management because of their low level of involvement in the process.

The systems test architect provides the test and evaluation activities a consistent voice throughout the life cycle of the system and enables the cooperation of testers and developers, leading to the early integration of test and evaluation input into development of the system. The systems test architect also helps translate test results into meaningful information to program managers, systems engineers and developers.

4. Risk Reduction

The systems test architect provides the program manager with another risk management tool; by virtue of development and execution of the test strategy the systems test architect addresses program and technical risks and program costs. As Bodmer (2003, 67) describes, "A disciplined and well-structured test program reduces the risk of acquiring an ineffective system and provides the program manager with timely information required to make prudent decisions during system development." The systems test architect addresses the risk of failing validation tests by virtue of the rigorous

test strategy planning and execution. The potential risks identified through test planning and execution are communicated and addressed in concert with program managers, system engineers and developers.

The systems test architect helps systems engineers and architects improve architecture definitions and complexities, during the process of developing the test strategy the systems test architect may uncover issues in the architecture than can be addressed much earlier, instead of being discovered during verification of validation tests much later in the development cycle.

5. Life-Cycle Cost Reduction

The impact of a rigorous test and evaluation program plan and execution should not only be evident in the short-term, when system requirements are validated and the system is deployed. But in the long term, over the entire life cycle, when the cost to sustain the deployed system is reduced by reduced maintenance needs, deficiency discoveries and ease of improvements. The systems test architect influences the system architecture to improve reliability, availability, maintainability, testability and interoperability aspects of the system early on.

The systems test architect enables cost reduction of the test and evaluation program by eliminating duplicative test efforts, utilizing modeling and simulation where warranted, early identification of defects and by coordinating events to wisely compress the test schedule.

B. A HYPOTHETICAL SCENARIO FOR THE SYSTEMS TEST ARCHITECT

The concept of systems test architect is not completely new, the model proposed in this thesis is different from any current documented models and as such there little of no data available to quantify the value added by the role. The scenario presented here is a contrasting view of a systems engineering task with and without a system test architect, showing the potential value of the role.

The Rush Company is developing a new system for a customer. Since Rush, Co. is a strong believer in systems engineering, it a major component of very development task. The company follows the simple but effective “Vee” process model.

The company leadership assign Mr. Lee as the program manager. Mr. Lee is an experienced program manager who has been through many systems engineering cycles during his career. For the duration of the development effort Mr. Lee decides to call the system product YYZ. As befitting any project initiation Mr. Lee begins to form his systems engineering team for the development effort.

At this point, let the reader envision two possible scenarios: one with and one without the systems test architect. These two scenarios demonstrate some of the pitfalls that a systems engineering team may encounter during a development effort. They also demonstrate how test and evaluation and the involvement of the systems test architect affects the avoidance of those pitfalls. As befitting the small scope hypothesis (Jackson 2012), these scenarios do not explore the entirety of the systems test architects’ roles, responsibilities and interactions. The scenarios presented here are just enough to demonstrate the potential value of the systems test architect to the systems engineering process.

Without the systems test architect [w/o]:

Although a veteran of systems engineering for product development, Mr. Lee is not a fan of test and evaluation activities and testers. It is his belief that they are a burden upon the program budget and a negative influence on the development schedule. Based on his past experiences, Mr. Lee decides to personally manage the entirety of the test and evaluation program. Mr. Lee has chosen Mr. Peart to serve as his system engineer and Mr. Rose to be his systems architect.

With the systems test architect [w]:

Having been through many systems engineering development cycle, Mr. Lee has become disenchanted with the frequency with which projects have been delayed by late-stage deficiency discoveries during test activities and the amount of associated rework.

He contends that his management skills are better suited for the overall program execution and that he would be well served by appointing a test and evaluation domain expert to take on the development and execution of the test strategy from day one. Mr. Lee has chosen Mr. Peart to serve as his system engineer and has appointed Mr. Sawyer to be his systems test architect and Mr. Rose to be his systems architect.

1. Definition of System Requirements

Mr. Lee and his systems engineering team hold a kickoff meeting with their customer and other stakeholders to establish the customer needs, high level requirements and constraints.

[w/o] The Mr. Peart, the systems engineer gleans the top level requirements for the system and begins to decompose those requirements into Measures of Efficiency and Measures of Performance. Unbeknownst to Mr. Peart and Mr. Lee, a requirement has been written which is incompatible with another requirement. Together with Mr. Lee, Mr. Pear begins to craft the systems engineering plan. Mr. Lee beings to craft the test and evaluation master plan.

[w] Mr. Peart, the systems engineer gleans the top level requirements for the systems and begins to decompose those requirements into Measures of Efficiency and Measures of Performance. Mr. Peart works with Mr. Sawyer, the systems test architect, to refine those requirements and measures based on their clarity, feasibility and testability. Mr. Sawyer knows the test and evaluation capabilities of the organization and observes that some of the requirements are questionable. Mr. Sawyer contacts the modeling and simulation group and requests an analysis of the requirement. Based on the results of their analysis, Mr. Sawyer is able to correct a requirement that would have been led to issues later in the development.

[w] Together with Mr. Lee, Mr. Peart and Mr. Sawyer begin to craft the systems engineering plan. Mr. Sawyer begins to craft the test and evaluation master plan in concert with the test group and requests the test and evaluation group for some estimates for test. He delivers these to Mr. Lee to ensure that enough funding has been set aside for verification and validation later in the cycle.

2. Functional Decomposition and Allocation

[w/o] Having reached a satisfactory level of requirements definition, Mr. Peart and Mr. Rose begin to decompose those requirements into functions and develops a functional architecture. The incompatible requirement has been passed down to the functional architecture and has resulted in a functional architecture with and complex interface.

[w] Having reached a satisfactory level of requirements definition, Mr. Peart and Mr. Rose begin to decompose those requirements into functions and develops a functional architecture. Having resolved the issue with the incompatible requirements, Mr. Rose concentrates on building an effective functional architecture but struggles with the interface complexities and asks Mr. Sawyer for assistance. Mr. Sawyer evaluates the interfaces and determines that one set of interfaces is divided into a large number of functions and recommends rearranging the architecture into some nested common groups.

3. Design Synthesis

[w/o] Mr. Peart and Mr. Rose take their functional architecture and create some alternate physical architectures, they work around the complex interface but finally arrive at what they judge as an acceptable physical architecture. They had some worries about the physical architecture, and had approached Mr. Lee for funds to perform some simulation studies. Mr. Lee, however, refused to make the funds available, insisting that they could figure out the issues during verification testing.

[w] Mr. Peart and Mr. Rose take their functional architecture and create some alternate physical architectures. Although satisfied that the architecture arrangement will meet the intended system goal, Mr. Sawyer engages the test and evaluation group to build and test some benchtop prototypes. This study revealed that another set of interfaces was too complex and would benefit from division into some nested components.

[w] Having a physical architecture in place, and having already developed some prototypes, Mr. Sawyer works with the testers to finalize the test plans for the upcoming verification phases.

4. Design Implementation

[w/o] Mr. Peart finally has a physical architecture, but is worried about the adequacy of the design. Grudgingly, Mr. Lee agrees to build a prototype of the system to work out some of the issues Mr. Peart feels are in the design. The builders have a hard time getting the prototype to perform its intended function, stating that there are two sets of functions that are incompatible. They ask Mr. Peart to reevaluate his architecture and intended functions. Mr. Lee is now furious, as he feels that the system would be fine as-is and it would require some "fine-tuning."

[w] Mr. Peart finally has a physical architecture, and feels confident that it will meet the desired systems function. Mr. Sawyer advises Mr. Peart that a prototype of the system would be prudent at this stage. Mr. Peart and Mr. Lee agree to have that prototype built, and the results was a minor change to the physical architecture. The system engineering team is confident enough to proceed with the verification stages.

5. Component Verification

[w/o] The builders and testers continue to create prototypes, engaging in several cycles of architecture changes,rebuilds and tests. Mr. Lee has become frustrated by the amount of funding spent by the test and evaluation group, but Mr. Peart has advised Mr. Lee that the system will not work as intended. Finally, Mr. Peart and the testers are able to convince Mr. Lee to change the top level requirements. This leads Mr. Peart and Mr. Rose toredo some of the functional and physical architectures.

[w] Mr. Peart realizes that at this stage, not test plans have been developed for each component and sub-assembly, and notifies Mr. Lee. Mr. Lee subsequently creates a very generic test plan that allows the system pass tests regardless of the actual outcome.

[w] The testers are able to test all the component and sub-assemblies without issues and concur with Mr. Peart and Mr. Sawyer that the system is ready to be assembled.

6. System Verification

[w/o] Having muddled through the component tests, the team finally has a completed system assembly. Using a cobbled together test plan, the test and evaluation

proceeds to test the system. However, they find that a lot of the test point, and data requirements are not defined. In concert with Mr. Peart, they settle on a data set for the test. But without a test and evaluation strategy, they are unsure if the results gathered will answer the question of whether the system was built right.

[w/o] In the end, the system was tested, and everything seemed to work properly.

[w] With a properly defined test and evaluation strategy, the tester is able to assemble the system and perform the tests. The well-defines data requirements have produced a set of data that allows Mr. Sawyer to evaluate the system and determine with statistical certainty that the system was built right.

7. System Validation

The finished system was given to the operational test group to validate. They determined that the system was meeting its intended function and no further development was required.

C. CHAPTER SUMMARY

This chapter presented the potential value of the role of the systems test architect in the systems engineering process. The value of the systems test architect emerges not just from the integration of test and evaluation earlier in the systems engineering process, but from the dedicated, constant presence that develops and executes the test strategy. The systems test architect improves the process of systems engineering from the earliest stages of requirements definition, but the greatest value that the systems test architect provides is communication between the development organization and test and evaluation workforce. Finally, a hypothetical scenario presented a possible set of outcome for a small development task, in both cases the same system was build, but with the systems test architect, the systems engineering team was able to find and resolve issues earlier in the development process.

The next chapter provides a short discussion of the research questions, followed by the conclusion drawn from this research effort and a recommendation on topics for further study.

VI. CONCLUSION AND RECOMMENDATIONS

A. DISCUSSION OF THE RESEARCH QUESTIONS

1. Where and how does the role of the systems test architect intersect with the other roles in the system engineering process, based upon their respective literary descriptions?

Like the systems architect, the systems test architect is a domain expert. The systems test architect is focused on the integration of test and evaluation in the systems engineering process. The systems test architect intersects the system engineering process across the entire process spectrum, from inception to delivery and beyond into operation and sustainment. The details of how the systems test architect intersects with systems engineering is detailed in the auxiliary questions.

a. What is the role of the systems test architect within the systems engineering process?

The primary role of the systems test architect is to be the systems thinker for test and evaluation. As detailed in the Chapter IV roles discussion, the systems test architect is the advisor to the decision-makers on matters of test and evaluation and the advocate for the test and evaluation process. The systems test architect is an enabler of communication across the process and organization and provides the domain-expert point of view to test requirements and results.

b. How does the systems test architect interact with the other roles within the systems engineering process?

The systems test architect is the primary integrator for the test and evaluation process. The Chapter IV interactions discussion shows that the systems test architect constantly communicates with program managers, systems engineers, developers, testers and evaluators to influence requirements for testability and ensures that testability is built into the design. The systems test architect helps testers develop their test plans and evaluates the results of tests, acting as the first-line filters for the data and providing feedback to the systems engineering team in the process.

c. How does the systems test architect improve the systems engineering process?

Chapter II details several ways that test and evaluation positively influences the systems engineering process, the systems test architect provides a bridge to those influences. The systems test architect ensures early, constant and consistent involvement of the test and evaluation workforce in the systems engineering process. The systems test architect provides a life cycle–focused level of attention to the test and evaluation process that can aid program reduce life-cycle costs. The systems test architect helps the program managers and systems engineers reduce risks associated with deficiencies and can help lower the cost of the test and evaluation program by taking a holistic view of testing. That is, the systems test architect can use their ownership of the process to combine test events, use modeling and simulation, and other design of experiments tools to reduce the resources required to execute the test architecture.

d. Should the role of the systems test architect be more thoroughly integrated (and formally defined) into systems engineering?

Based on the potential value exposed in Chapter V, the systems test architect should become an integral part of the systems engineering process. Systems engineering focuses mainly on the tools and techniques that add value and increase the probability of success. The people that choose and exercise those tools are just as important to the process. The systems test architect is part of the framework of interdisciplinary professionals that give the systems engineering approach its value.

e. How does the systems test architect affect the cost and quality of a project?

Like any additional personnel, the systems test architect will incur a cost upon the program. It is the program managers' imperative to weigh the cost of an additional body in the program structure against the potential financial benefits discussed in this research and explored briefly in the hypothetical scenario of Chapter VI. The program manager must consider two strong arguments: first is that someone has to do the work of integrating test and evaluation, as testing is an indispensable activity throughout the systems engineering process. Second is that the potential costs and time required to

address deficiencies in the systems requirements, design, or other attributes would outweigh the costs associated with the extra personnel.

B. CONCLUSIONS

It is an undeniable truth that test and evaluation activities are required and value-added steps in any system development or acquisition. The question any systems engineer should pose themselves is how best to integrate test and evaluation in their process. The systems test architect gives the systems engineer an answer to that question.

Test and evaluation professionals like the systems test architect are part of the framework that systems engineering needs in order to fulfill the requirements of the system and ensure mission success. Systems test architects are part of the equation in implementing the “shift-left” mentality in the DOD and in system engineering in general.

Ultimately, the systems test architect is no silver bullet; there are no silver bullets in systems engineering. Mainly because systems engineering in both technical and managerial, where no one approach will repeatedly achieve the same end result. No one approach, model or process will prevent all defects in the system from emerging, but they can be reduced. And no role in the process can affect the process without buy-in from the holders of the purse-strings. The systems test architect is another piece in the framework of systems engineering focused professionals that add value and increase the probability of success of the development effort.

DOD recognizes the value of test and evaluation, as evident in Title 10 and DOD 5000; however, no similar guidance is available directed at the general system engineering process. The establishment of a role like the systems test architect allows the systems engineering process to reap the benefits of a rigorous test and evaluation program.

Early involvement of test and evaluation in systems engineering requires a concerted effort, as it stands, program managers and systems engineers are responsible for addressing test and evaluation. The integration of a role like the systems test architect allows for a domain expert to take on the responsibility to make this involvement a

reality. However, this can only happen if the organizational structure provides its concurrence and support, and there must exist an environment that allows trust, respect, open communication and collaboration within the organization that allows both the individual and team thrive and succeed,

By viewing test and evaluation as a stakeholder, the focus shifts to satisfying the needs of test and evaluation. The systems test architect gives test and evaluation a voice and influence over the process in which they participate that would otherwise be easily silenced and is currently diluted.

The available guidance on the involvement of test and evaluation is either incomplete, vague or too focused on a specific context. Someone has to do the job, someone has to work on and pursue the details on how/where/when each requirement is tested. Most textbooks and guides are focused on the technical aspects of systems engineering, very few sources give the managerial aspect of system engineering any attention. DOD guidance is well rounded in that it details both technical and managerial aspect of the acquisition process, while industry is lagging behind by not addressing the multi-disciplinary needs of the management of the systems engineering process.

In the research for this topic, the only company with a published record of implementing a role responsible for the test and evaluation process is Raytheon Missile Systems. Their presentation at NDIA conferences were the seed of the authors' formulation of the systems test architect model in this research. If one major defense contractor found the value in this role, there is no reason why other industry bodies utilizing systems engineering will not benefit as well.

Although test and evaluation is time consuming and can account for a good portion of the investment into a program development effort, a rigorous, well-structured test and evaluation program can and should result in risk and cost reductions.

C. TOPICS FOR FUTURE WORK

This research was born out of the author's experience as a test and evaluation workforce member. The drive behind this research was an observed pattern related to the

lack of involvement of the test personnel and resources during development efforts. These efforts would seek to utilize test resources without any plan in place and without forethought to test requirements. During the development of the topic for this research, the concept of the systems test architect was discovered and adopted as a topic for further study, based on the lack of documentation about that role. The following topics could provide further insight into this role and perhaps other that are missing in the management of the systems engineering process:

- Collect and analyze case studies surrounding the role of the systems test architect. These case studies may gauge the effectiveness of the role.

- Compare and contrast the effectiveness of the centralized role of the systems test architect to a decentralized integrated product team. This sets the threshold for the program manager to decide what level of effort requires a systems test architect and what level would benefit from an integrated product team.

- Expand the hypothetical scenario in this thesis using business processing modeling tools.

- Determine what other areas of systems engineering are underdeveloped regarding the roles that carry out the technical processes. Is system engineering a purely technical process, or it is both technical and managerial?

- Analyze model-based systems engineering tools for test and evaluation support.

- Create and/or integrate a conceptual data model for test and evaluation integration into model-based systems engineering.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Armed Forces, 10 U.S.C., Subtitle A §139b (2015).

Army Test and Evaluation Command. 2003. “Test and Evaluation in Support of Systems Acquisition.” Washington, D.C.: Army Test and Evaluation Command.

Barret, Christopher J. 2009. “Integrating Test and Evaluation into the Acquisition Process for Naval Aviation.” Master’s thesis, Naval Postgraduate School, Monterey, CA.

Blanchard, Benjamin S., and Wolter J. Fabrycky. 2011. *Systems Engineering and Analysis*. Upper Saddle River, NJ: Prentice Hall.

Bodmer, Gerald A. “Test and Evaluation in the United States Navy, and How it Must Evolve to Support Future Systems Acquisition.” Thesis, Naval Postgraduate School, Monterey, CA, 2003.

Brewer, James, Louisa Guise, and Jerry Emmert. 2012. “The Value of the Test Architect.” Paper presented at the NDIA 28th Annual National Test and Evaluation Conference, Hilton Head, SC, March 20.

Buede, Dennis M. 2009. *The Engineering Design of Systems*. Hoboken, NJ: John Wiley & Sons, Inc.

Defense Acquisition University. 2001. *Systems Engineering Fundamentals*. Fort Belvoir, VA: Defense Acquisition University Press.

———. 2012. *Test and Evaluation Management Guide*. Fort Belvoir, VA: Defense Acquisition University Press.

Grady, Jeffrey O. 2010. *System Synthesis*. San Diego, CA: CRC Press.

Guise, Louisa. 2012. “Systems Engineering: Providing the Right Testing at the Right Time.” Paper presented at the 28th Annual National Test & Evaluation Conference. Hilton Head, SC, March 14.

International Council on Systems Engineering (INCOSE). 2016a. “About INCOSE.” May 31. http://www.incose.org/about.

———. 2016b. “What is Systems Engineering?” May 26. http://www.incose.org/AboutSE/WhatIsSE.

Jackson, Daniel. 2012. *Software Abstractions: Logic, Language, and Analysis*. Cambridge, MA: MIT Press.

Maier, Mark W., and Eberhardt Rechtin. 2009. *The Art of Systems Architecting*. Boca Raton, FL: CRC Press.

Manas, Joe. 2014. "The Chief Developmental Tester and Industry Test Lead – Partnering for Success." Paper presented at the NDIA 17th Annual Systems Engineering Conference. Springfield, VA, October 30.

Manas, Joe, and Beth Wilson. 2014. "Test Perspectives for Architecture." Paper presented at the 17th Annual Systems Engineering Conference. Springfield, VA, October 30.

Merriam-Webster. 2016. "Architect." May 26. http://www.merriam-webster.com/dictionary/architect.

Morrison, John. 2007. "Test Architect." John Morrison's Blog. https://blogs.oracle.com/johnmorrison/entry/test_architect.

Page, Alan. 2008. "What Is a Test Architect?" *Notes And Rants* (blog). https://blogs.msdn.microsoft.com/alanpa/2008/05/01/what-is-a-test-architect.

Roedler, Garry J., and Cheryl Jones. 2005. "Technical Measurement: A Collaborative Project of PSM, INCOSE, and Industry." INCOSE-TP-2003-020-01. Army Armament Research Development and Engineering Center Picatinny Arsenal NJ.

Science Applications International Corporation. 2002. "A Study of Commercial Industry Best Practices in Test & Evaluation which are Potentially Applicable to DOD Developmental Test and Evaluation." Washington, D.C.: Deputy Director, Developmental Test and Evaluation Office of the Undersecretary of Defense (Acquisition, Technology and Logistics).

Under Secretary of Defense for Acquisition, Technology, and Logistics. 2003. *The Defense Acquisition System*. DOD Directive 5000.01. Washington, D.C.: Under Secretary of Defense for Acquisition, Technology, and Logistics. http://www.dtic.mil/whs/directives/corres/pdf/500001p.pdf.

———. 2015. *The Defense Acquisition System.* DOD Directive 5000.01. Washington, D.C.: Under Secretary of Defense for Acquisition, Technology, and Logistics. http://www.dtic.mil/whs/directives/corres/pdf/500002p.pdf.

United States Department of Defense. 2014. *Defense Acquisition Guidebook*. Washington, D.C.: Defense Acquisition University.

United States General Accountability Office. 2000. *Best Practices: A More Constructive Test Approach Is Key to Better Weapon System Outcomes.* NSIAD-00-199. Washington, D.C.: National Security and International Affairs Division (GAO).

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California